

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL**

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE VISIÓN
ESTEREOSCÓPICA PARA EL POSICIONAMIENTO DEL BRAZO
ROBÓTICO CRS-A255”**

AUTORES:

GUSTAVO ESTEBAN CHÁVEZ MORALES

DANIEL ALEJANDRO FLORES PARRA

SANGOLQUÍ – ECUADOR

2011

CERTIFICACIÓN

Certificamos que el presente proyecto titulado:

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE VISIÓN ESTEREOSCÓPICA PARA EL POSICIONAMIENTO DEL BRAZO ROBÓTICO CRS-A255”

Ha sido desarrollado en su totalidad por los señores GUSTAVO ESTEBAN CHÁVEZ MORALES Y DANIEL ALEJANDRO FLORES PARRA bajo nuestra dirección.

Ing. Paul Bernal

DIRECTOR

Ing. Rodolfo Gordillo

CODIRECTOR

AGRADECIMIENTO

Agradezco a Dios, a mi familia, a la Escuela Politécnica del Ejército, a todos quienes han sido mis profesores especialmente a mis directores de tesis Luis Orozco, Julio Larco, Paul Bernal y Rodolfo Gordillo, a mi colega en la investigación Daniel Flores, a todos mis amigos, especialmente a Marcelo, a María y a mi entrañable aliada Marcela.

Esteban Chávez

Dejo constancia de mi eterno agradecimiento a Dios sobre todo, por hacer posible este proyecto, luego a Gonzalo Parra, mi madre Geanneth Parra y toda mi familia por estar siempre a mi lado, a mi colega de investigación Esteban Chávez, a la Escuela Politécnica del Ejército, a mis profesores por sus consejos y guías y por último a Gabriela Montenegro y todos mis amigos por siempre darme apoyo y compañía.

Daniel Flores

DEDICATORIA

A mis padres, Gustavo y Martha.

Esteban Chávez

Dedico este proyecto a mi madre, mi familia y amigos. Así como también a la distinguida Escuela Politécnica del Ejercito y su departamento de Eléctrica y Electrónica y por último a todas las personas que piensen que en el Ecuador si es posible la generación de tecnología.

Daniel Flores

PRÓLOGO

Desde que el hombre existe, una de sus mayores ambiciones ha sido reproducir los sistemas biológicos que lo conforman, fundamentalmente aquellos que han sido llamados sentidos, tacto, olfato, gusto, oído y visión. La agrupación de estos sentidos en un organismo que actúe con inteligencia, capaz de realizar tareas y bajo la voluntad del hombre toma el nombre de androide.

La visión es el más importante de los sentidos, al recibir aproximadamente el ochenta por ciento de la información sensorial que llega al cerebro. El hombre ya ha sido capaz de capturar su entorno simulando así una de las primeras funciones que tiene la visión, sin embargo este sentido realiza numerosos procesos que son estudiados con vehemencia en la actualidad. El reconocimiento de formas, estimación de profundidad, tamaño, color, textura y trayectoria de objetos son solo algunos de estos procesos.

Al fijar nuestra atención en la estimación de la profundidad aparece de inmediato la técnica de visión estéreo. Esta toma como base el modelo biológico, al tener dos ojos podemos percibir la profundidad de los objetos gracias a la diferencia entre las dos imágenes, y el proceso implícito que se realiza en el cerebro.

En este contexto, el presente proyecto implementa un sistema de visión computacional que mediante la técnica estereoscópica calcula la profundidad y ubicación de un objeto y trasmite esta información al sistema robótico CRS-A255 para que se posicione sucesivamente.

ÍNDICE

CAPÍTULO 1: MARCO TEÓRICO

1.1 Visión Computacional.....	1
1.1.1 Concepto	1
1.1.2 Imágenes.....	5
1.1.3 El Color.....	6
1.1.3.1 Modelos de color normalizados	8
1.1.4 Representación de imágenes.....	11
1.1.4.1 Representación basada en color	11
1.1.4.2 Representación basada en textura	12
1.1.4.3 Representación basada en forma	13
1.2 Sistemas de Visión Computacional	14
1.2.1 Concepto	14
1.2.2 Sistema de Visión estereoscópica.....	14
1.2.2.1 Percepción de la Profundidad.....	14
1.2.3 Arquitectura de un Sistema de Visión Computacional.....	15
1.2.3.1 Dispositivo de Captura.....	15
1.2.3.2 Conversión A/D.....	15
1.2.3.3 Memoria de Vídeo.....	16
1.2.3.4 Procesador	16
1.2.3.5 Monitor.....	16
1.2.4 Niveles de análisis.....	16
1.2.5 Niveles de visión	17
1.3 Visión Estereoscópica	18

1.3.1 Definición y Principios	18
1.3.2 Geometría del sistema estéreo y principio de obtención de la distancia	19
1.3.2.1 Geometría de ejes paralelos	20
1.3.3 Algoritmos de visión estéreo o el problema de la correspondencia.....	23
1.3.3.1 Restricciones Aplicadas al problema de correspondencia.....	23
1.3.3.2 Clasificación de algoritmos de correspondencia.....	24
1.3.4 Consideraciones de sistemas de visión estéreo.....	27
1.3.4.1 Tiempo Real en Visión Estéreo.....	27
1.3.4.2 Velocidad o FPS de un sistema estéreo	27
1.4 Sistema Robótico Industrial CRS A-255.....	28
1.4.1 Brazo robótico articulado.....	28
1.4.1.1 Sistema Coordinado del Mundo.....	29
1.4.1.2 Sistema Coordinado de la Herramienta.....	30
1.4.1.3 Sistema Coordinado Cilíndrico	31
1.4.1.4 Sistema de Coordenadas de las articulaciones	32
1.4.2 Controlador robótico C500	33
1.4.2.1 Interfaz de operación	33
1.4.2.2 Comunicación	34

CAPÍTULO 2: SISTEMA DE VISIÓN ESTEREOSCÓPICA

2.1 Introducción.....	37
2.1.1 Variables a obtenerse y características.....	37
2.1.2 Requerimientos de operación del sistema.....	38
2.1.2.1 Condiciones de trabajo	39
2.1.2.2 Condiciones de Usuario.....	39
2.1.3 Limitantes inherentes al proyecto.....	39
2.2 Selección del algoritmo de correspondencia.....	40

2.2.1 Costo Computacional	41
2.2.2 Versatilidad.....	41
2.2.3 Relación con la tarea del proyecto	42
2.2.4 Requisitos de utilización y rentabilidad.....	42
2.2.5 Algoritmo de correspondencia seleccionado	43
2.3 Algoritmo de ubicación en el plano imagen.....	43
2.3.1 Variable longitud – pixel	45
2.3.2 Método de característica conocida.....	45
2.3.2.1 Ventajas.....	46
2.3.2.2 Desventajas	46
2.3.3 Método de Aproximación por distancia.....	47
2.3.3.1 Ventajas.....	48
2.3.3.2 Desventajas	49
2.3.4 Selección del algoritmo de ubicación en el plano imagen	49
2.4 Arquitectura Del Sistema De Visión	49
2.4.1 Diagrama en bloques de elementos.....	49
2.4.1.1 Dispositivos de Captura	51
2.4.1.2 Unidad central de Proceso.....	51
2.4.1.3 Video.....	52
2.5 Algoritmo Implementado	52
2.5.1 Diagrama de Flujo	53
2.5.2 Adquisición de imágenes.....	54
2.5.3 Procesamiento de la imagen	57
2.5.4 Detección de características	59
2.5.5 Extracción de información	61

CAPÍTULO 3: ALGORITMO DE POSICIONAMIENTO

3.1 Introducción.....	65
3.2 Comunicación e Inicialización	66
3.2.1 Comunicación.....	66
3.2.2 Inicialización	66
3.3 Descripción del Problema de Movimiento	71
3.3.1 Escenario previo.....	71
3.3.2 Problema de Orientación.....	72
3.3.3 Descripción de la estrategia propuesta	74
3.3.4 Comandos a utilizarse	75
3.4 Algoritmo Implementado	75
3.4.1 Inicialización	77
3.4.2 Condiciones de Funcionamiento	80
3.4.3 Movimiento del Robot.....	83

CAPÍTULO 4: PRUEBAS Y RESULTADOS

4.1 Introducción.....	86
4.2 Métodos	86
4.2.1 Obtención de la distancia.	86
4.2.1.1 Descripción	86
4.2.1.2 Resultados.....	90
4.2.2 Obtención de la posición en el plano.....	91
4.2.2.1 Descripción	91
4.2.2.2 Resultados.....	93
4.3 Pruebas.....	93
4.3.1 Color del objeto	93
4.3.2 Deformación del objeto.....	95

4.3.3 Tamaño del objeto.....	96
4.3.4 Objetos múltiples.....	97
4.3.5 Volumen de trabajo	98
4.3.6 Velocidad del Manipulador	99
4.3.7 Fondo del Objeto.....	99
4.3.8 Niveles de iluminación.....	101

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones.....	104
5.2 Recomendaciones.....	107

ANEXOS

ANEXO A: SISTEMA DE VISIÓN ESTEREOSCÓPICA COMERCIAL

ANEXO B: COMANDOS DE MOVIMIENTO RAPL-II

ANEXO C: MÉTODOS EXPERIMENTALES

CAPÍTULO 1

MARCO TEÓRICO

1.1 VISIÓN COMPUTACIONAL

1.1.1 Concepto

Visión es la ventana al mundo de muchos organismos. Su función principal es reconocer y localizar objetos en el ambiente mediante el procesamiento de las imágenes. La visión computacional es el estudio de estos procesos, para entenderlos y construir maquinas con capacidades similares [1]. Existen varias definiciones de visión, entre estas podemos mencionar las siguientes:

- “Visión es saber que hay y donde mediante la vista”, (Aristóteles).
- “Visión es recuperar de la información de los sentidos (vista) propiedades validas del mundo exterior”, (Gibson).
- “Proceso que a partir de imágenes del mundo externo produce una representación útil al observador y no está ensombrecida por información irrelevante”, (David Marr).

Las tres son esencialmente validas, pero la que tal vez se acerca más a la idea actual sobre visión computacional es la definición de Marr. En esta definición hay tres aspectos importantes que hay que tener presentes: (i) visión es un proceso computacional, (ii) la descripción a obtener depende del observador y (iii) es necesario eliminar la información que no sea útil (reducción de información).

Un área muy ligada a la de visión computacional es la de procesamiento de imágenes. Aunque ambos campos tiene mucho en común, el objetivo es diferente. El objetivo del procesamiento de imágenes es mejorar la calidad de las imágenes para su posterior utilización o interpolación, por ejemplo:

- Remover defectos
- Remover problemas por movimiento o desenfoque
- Mejorar ciertas propiedades como color, contraste, estructura, etc.
- Agregar “colores falsos” a imágenes monocromáticas.

En la figura 1.1 se ilustra el enfoque de un procesamiento de imágenes, en el cual se obtiene una imagen “mejor” para su posterior interpretación por una persona.

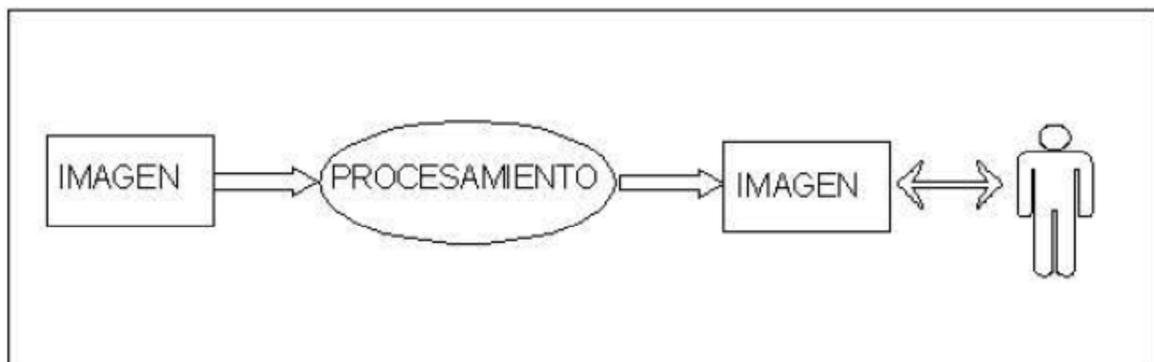


Figura. 1.1. Esquema general del procesamiento de imágenes.

Su función principal es presentar la misma imagen resaltando e ignorando ciertas características. Obsérvese que la entrada y salida son imágenes.

El objetivo de la visión computacional es extraer características de una imagen para su descripción e interpretación por la computadora. Por ejemplo:

- determinar la localización y tipo de objetos en la imagen.
- construir una representación tridimensional de un objeto.
- analizar un objeto para determinar su calidad.
- descomponer una imagen u objeto en diferentes partes.

En visión se busca obtener descripciones útiles para cada tarea a realizar. La tarea demandara modificar ciertos atributos, ver figura 1.2.

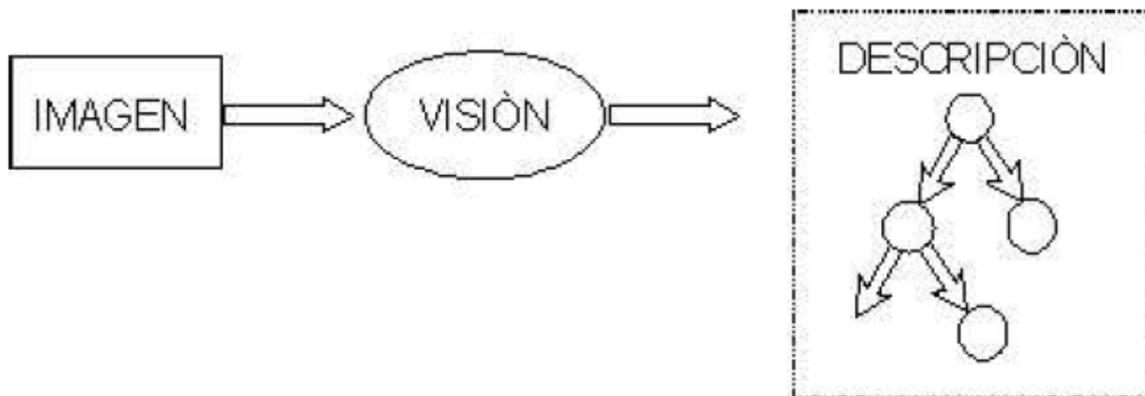


Figura. 1.2. Esquema general de visión por computadora.

La imagen de entrada es procesada para extraer los atributos, obteniendo como salida una descripción de la imagen analizada.

En la figura 1.3 se muestra un ejemplo de procesamiento de imágenes. La tarea a realizar es mejorar la imagen de entrada, la cual es oscura (a). La imagen de salida (b) es esencialmente la misma pero de mejor calidad o “más útil”.



Figura. 1.3. Ejemplo procesamiento de imágenes.

La figura 1.4 ilustra la diferencia entre procesamiento de imágenes y visión; nótese que la imagen muestra ciertas descripciones importantes, como los números, que previamente fueron detectados. La salida de este sistema de visión se complementa con un modulo de reconocimiento de patrones, es decir, “saber” qué letras contiene la placa.

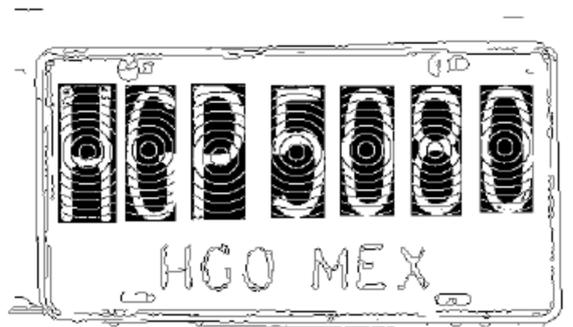


Figura. 1.4. Ejemplo sistema de visión.

Actualmente existen múltiples aplicaciones prácticas de la visión computacional, entre estas podemos mencionar las siguientes:

- Robótica móvil y vehículos autónomos. Se utilizan cámaras y otros tipos de sensores para localizar, identificar objetos y personas, encontrar el camino, etc.
- Manufactura. Se aplica visión para localizar e identificación de piezas, para control de calidad, entre otras tareas.
- Interpretación de imágenes aéreas y de satélite. Se usa procesamiento de imágenes y visión para mejorar las imágenes obtenidas, para identificar tipos de cultivos, para ayudar en la predicción del clima, etc.
- Análisis e interpretación de imágenes médicas. La visión se aplica para ayudar en interpretación de diferentes clases de imágenes médicas como rayos-X, tomografía, ultrasonido, resonancia magnética y endoscopia.
- Interpretación de escritura, dibujos, planos. Se utilizan técnicas de visión para el reconocimiento de textos, lo que se conoce como reconocimiento de caracteres. También se aplica a la interpretación automática de dibujos y mapas.
- Análisis de imágenes microscópicas. El procesamiento de imágenes y visión se utilizan para ayudar a interpretar imágenes microscópicas en química, física y biología.
- Análisis de imágenes para astronomía. Se usa la visión para procesar imágenes obtenidas por telescopios, ayudando a la localización e identificación de objetos en el espacio.
- Análisis de imágenes para compresión. Aunque la compresión de imágenes ha sido tradicionalmente una sub área del procesamiento de imágenes, recientemente se están desarrollando técnicas más sofisticadas de compresión que se basan en la interpretación de las imágenes.

1.1.2 Imágenes

La Imagen bidimensional es una función que a cada par de coordenadas (x, y) asocia un valor relativo a alguna propiedad del punto que representa (por

ejemplo su brillo o su matiz). Una imagen acromática, sin información de color, en la que a cada punto se le asocia información relativa al brillo, se puede representar como una superficie (ver figura 1.5), en la cual la altura de cada punto indica su nivel de brillo. Una imagen en color RGB se puede representar asociando a cada punto una terna de valores que indica la intensidad de tres linternas (una roja, otra verde y otra azul). Una imagen de color de espectro completo se puede representar asociando a cada punto un diagrama espectral de emisión de color.

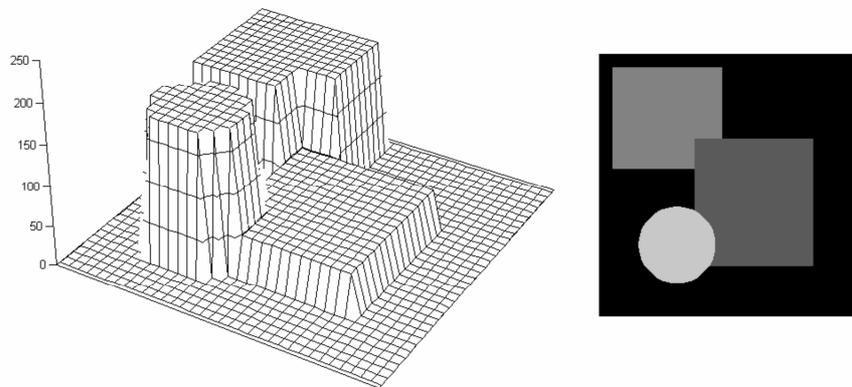


Figura. 1.5. Imagen 2D representada en 3D.

1.1.3 El Color

El Color es un elemento fundamental en los gráficos por computadora, por cuanto tiene una gran importancia en la calidad de visualización de una imagen en pantalla, ya que de él depende el resultado final de cualquier programa gráfico.

La percepción visual de un objeto ya sea en la propia realidad o a través de una pantalla de ordenador, es una percepción subjetiva, pues depende del propio ojo humano como órgano fisiológico (del que ya se pueden medir algunos de sus parámetros de sensibilidad) y del cerebro como órgano psico-fisiológico (del que aún no se pueden parametrizar sus respuestas con exactitud).

El fenómeno completo de la percepción visual implica una extensión, una duración y un color, que es el que nos definirá básicamente el objeto a partir de una luminosidad o claridad, un tono y una saturación o nivel de intensidad.

Así pues el concepto de color puede considerarse como una percepción subjetiva de visualización, pues aparte de las características intrínsecas del objeto y del tipo de iluminación a que esté sometido, depende fundamentalmente de las características fisiológicas de la persona que observa.

Aunque no se puede considerar al color como un parámetro estándar y medible objetivamente, desde siempre se ha pretendido teorizar sus características para establecer unos criterios de medición, que han dado como resultado la aceptación de unas recomendaciones CIE “*Commission Internationale de L’Eclairage*”, que unifican criterios y establecen unos parámetros básicos para su medición y tabulación.

En el estudio del color se han definido unos cuantos atributos que son los que se intentarán cuantificar para una identificación normalizada del mismo.

- **Energía luminosa o luz.** Es la parte de energía visible que incide directamente en la retina del ojo.
- **Color.** Entendido como color percibido es la diferenciación entre dos objetos en función del espectro de la energía radiante.
- **Tono.** Es la propia percepción del color tal como se le conoce coloquialmente (verde, rojo, etc.).
- **Brillo.** Sensación que se produce de colores más o menos intensos según se emita mayor o menor cantidad de luz (las variaciones de brillo van de brillante a tenue).
- **Luminosidad o claridad.** Sensación de mayor o menor luz emitida con relación a un estímulo blanco (sus variaciones van de luminoso a oscuro).
- **Cromaticidad o croma.** Sensación que diferencia un estímulo cromático, de otro acromático distinto, pero del mismo brillo, perceptible por la conjunción del tono y la saturación.
- **Saturación.** Sensación que diferencia un estímulo cromático de uno acromático independientemente de su brillo.

1.1.3.1 Modelos de color normalizados

Existen muchos modelos de color normalizados que entre los más utilizados se puede nombrar los siguientes:

Modelo RGB. Este modelo se puede considerar como una transformación lineal del modelo triestimulo XYZ, simplificando dichas funciones a tres variables mas sensitivas como son los porcentajes de rojo, verde y azul.

Estas tres variables RGB (rojo, verde, azul), forman un modelo tridimensional en forma de cubo (ver figura 1.6) basado en la teoría tricromática de la visión de color.

Este modelo está basado en la mezcla aditiva de los tres colores primarios para obtener cualquier color; es decir para producir un determinado color se suman las contribuciones individuales de cada primario.

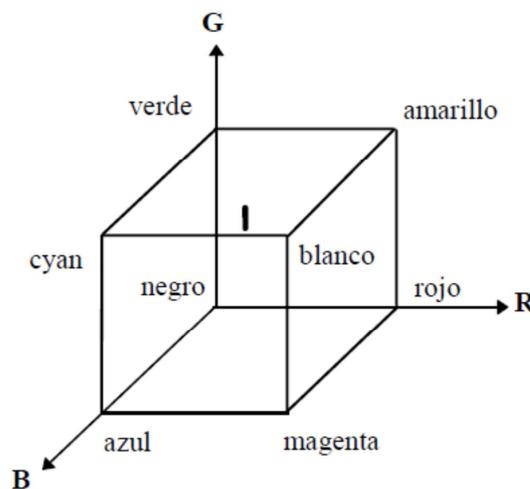


Figura. 1.6. Modelo RGB

Modelo CMY (CMYK). Este modelo es el complementario del anterior RGB considerando tres primarios sustractivos Cian Magenta y Amarillo en cuya ausencia se obtiene el color blanco tal como se aprecia en la figura 1.7

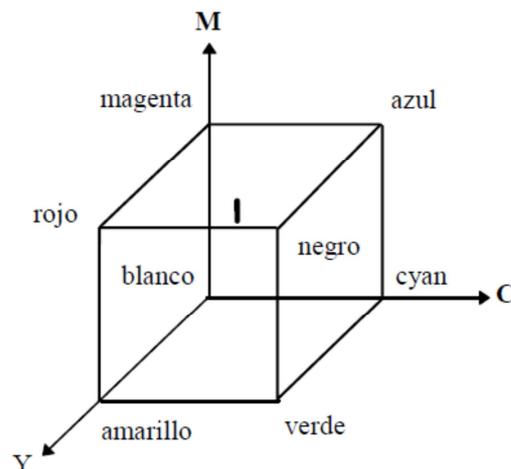


Figura. 1.7. Modelo CMY

En la figura 1.8 se puede apreciar la diferencia entre los dos modelos relativamente parecidos

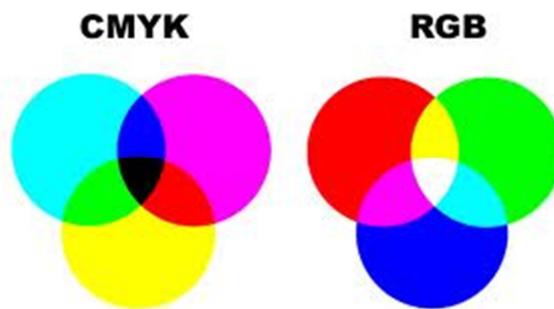


Figura. 1.8. Modelos CMYK y RGB

Modelo HSV. Este modelo aplica la descripción del color en los términos de Tono (H), Saturación (S) y Valor de brillantez (V). En este modelo el sistema de coordenadas es cilíndrico y el espacio de color queda determinado por una pirámide hexagonal tal como se ve en la figura 1.9.

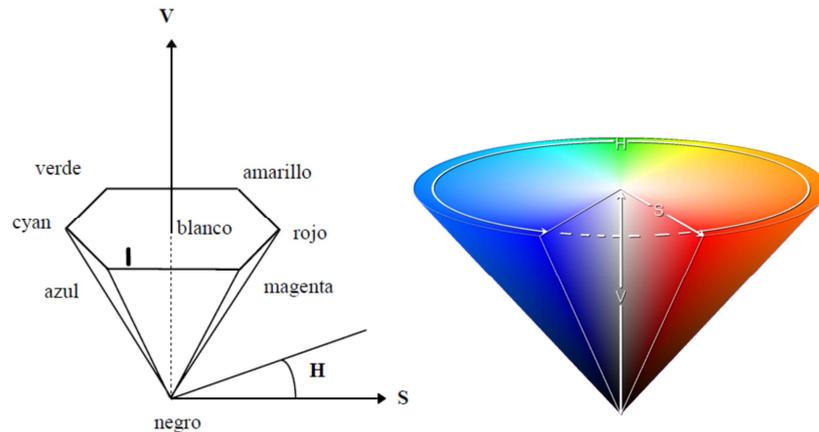


Figura. 1.9. Modelo HSV

Este modelo también tiene una analogía con el modelo, RGB ya que al observar el cubo según una proyección que se transforme en hexágono, este coincide con la base de la pirámide hexagonal. La parte superior de la pirámide hexagonal corresponde a la proyección que se puede ver a lo largo de la diagonal principal del cubo RGB. Este modelo se adapta perfectamente al trabajo de tintas, sombras y tonos.

Modelo HLS. Este modelo similar al anterior utilizando las variables de Tono (H), Luminosidad (L) y Saturación (S), en el que se ha sustituido la V por L.

En este modelo se percibe la sensación de más oscuro o más claro en lugar del blanco, dando lugar a una doble pirámide hexagonal.

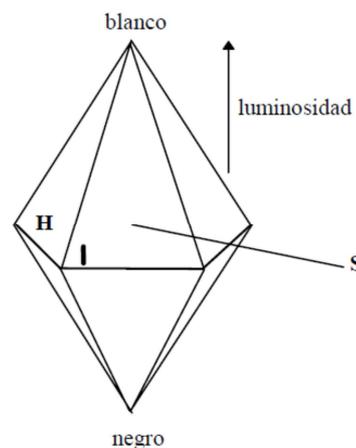


Figura. 1.10. Modelo HLS

1.1.4 Representación de imágenes

Esta parte está dedicada a explicar varias técnicas que sirven para obtener información relevante a las imágenes, dichas técnicas son representaciones que se basan sobre alguna sus características principales. Las representaciones que se verán son: basadas en color, basadas en textura, basadas en forma, basadas en ontologías y basadas en agentes de Software.

1.1.4.1 Representación basada en color

En sistemas de recuperación visual la información que nos brinda el color ha sido usada como una herramienta poderosa para dichos sistemas. Un ejemplo de un sistema que utiliza métodos basados en color es “*PicToSeek*” [8]. Este sistema fue desarrollado por la Universidad de Ámsterdam.

Para el propósito de recuperar información basada en color, se define una taxonomía de acuerdo al criterio siguiente: [9]

- Uniformidad perceptual
- Linealidades
- Intuitivita
- Robustez en contra de las condiciones variantes de las imágenes
- Invariancia al cambio de la vista de dirección
- Invariancia al cambio de la geometría del objeto
- Invariancia al cambio de la dirección de la iluminación
- Invariancia al cambio de la intensidad de la iluminación
- Invariancia al cambio de la distribución del poder espectral de la iluminación.

Esta taxonomía puede ser usada para seleccionar el sistema de color apropiado para una aplicación específica.

1.1.4.2 Representación basada en textura

Junto con el color, la textura es una poderosa característica discriminadora, presente en casi cualquier parte en la naturaleza. La textura puede ser descrita de acuerdo a su espacio, frecuencia o propiedades preceptuales. Periodicidad, aspersionidad, dirección, grado de complejidad son algunos de los más preceptuales atributos de la textura.

Características espaciales basadas en estos atributos son particularmente interesantes para recuperar imágenes por similitud de textura. Como los colores, la textura es conectada con efectos psicológicos.

Casi es imposible describir la textura por medio de palabras. Una efectiva descripción de textura puede estar basada en propiedades estadísticas y estructurales de patrones del brillo. Algunos investigadores han usado alguna de estas propiedades para buscar y recuperar. Existen muchas técnicas para extraer descriptores de textura significante de las imágenes, basadas en diferentes modelos y asunciones. Estos métodos se pueden clasificar de la siguiente manera:

- Modelos basados en espacio
- Función de auto-correlación
- Matriz de co-ocurrencia
- Dimensión fractal
- Modelos auto-regresivos
- Modelos estocásticos
- Modelos basados en frecuencia
- Poder espectro
- Transformación de Wavelets
- Firmas de textura
- Energía textual
- Contraste, aspersionidad y direccionamiento
- Repetición, direccionamiento, complejidad

Algunos ejemplos de sistemas que usan textura como principal descriptor para recuperar imágenes son *Virage Engine*, sistema CANDID y el sistema NeTra.

1.1.4.3 Representación basada en forma

No hay una definición universal de lo que es forma, pero es posible mencionar algunas definiciones bien aceptadas. Forma es la manera en que se ve un objeto definido por sus límites; forma es la apariencia externa de algo [9].

En general estos tipos de representaciones usan lo que se denomina *Shape Matching* (Correspondencia de Formas) el cual compara formas y es considerado como uno de los aspectos más difíciles en la recuperación de información visual basada en contenido, ya que la representación de formas frecuentemente es más compleja que el color y la textura, esto se debe porque una forma común necesita muchos parámetros para ser representados explícitamente.

Determinar la similitud de dos formas es una tarea muy difícil en sistemas tanto humanos como de máquina. Procesamiento de imágenes y visión computacional han proveído diferentes formas de representar imágenes así como también diferentes medidas para calcular qué tan similar es una imagen de otra.

1.2 SISTEMAS DE VISIÓN COMPUTACIONAL

1.2.1 Concepto

Un sistema de visión computacional es un conjunto de elementos que permite realizar tres funciones básicas: obtener imágenes del entorno, procesar estas imágenes para extraer características de interés y tomar decisiones que se basan en la evaluación de las características de las imágenes.

Las características de interés que se podrían extraer de las imágenes capturadas son numerosas y dependiendo de ellas, varían los elementos que conforman un sistema de visión. Por ejemplo si se quiere extraer la característica de profundidad de objetos de una imagen usando la técnica estereoscópica, se deberá utilizar dos cámaras para la adquisición de imágenes en contraposición con la mayoría de sistemas que solo usan una cámara. Sin embargo todo sistema de visión cumple con una arquitectura básica independiente de la característica a extraer de las imágenes

1.2.2 Sistema de Visión estereoscópica

Un sistema de visión estereoscópica es un tipo específico de sistema de visión. Como ya se menciona un sistema de visión, tiene por finalidad otorgar valores de diferentes características de una imagen, por ejemplo, color, forma, luminosidad, etc. En este sentido la principal finalidad de un sistema de visión estereoscópica es otorgar valores de la característica llamada profundidad en una escena de interés.

1.2.2.1 Percepción de la Profundidad

Es la habilidad visual de percibir el mundo en tres dimensiones. Es un rasgo común de muchos animales superiores. La percepción de la profundidad permite al que la tiene medir con precisión la distancia hasta un objeto.

1.2.3 Arquitectura de un Sistema de Visión Computacional

Un sistema de visión típico cuenta al menos con cinco elementos básicos, la disposición de estos elementos y el flujo de información entre sí conforman la arquitectura de un sistema de visión, ver figura 1.12, estos elementos son:

1.2.3.1 Dispositivo de Captura

Dispositivo físico que es sensible a una determinada banda del espectro electromagnético. El dispositivo produce una señal eléctrica proporcional al nivel de energía detectado. Entre los dispositivos de captura podemos mencionar:

- Imagen (CCD, CMOS)
- Infrarrojo (InGaAs refrigerado, HgCdTe)
- Rayos X, Luz Ultravioleta

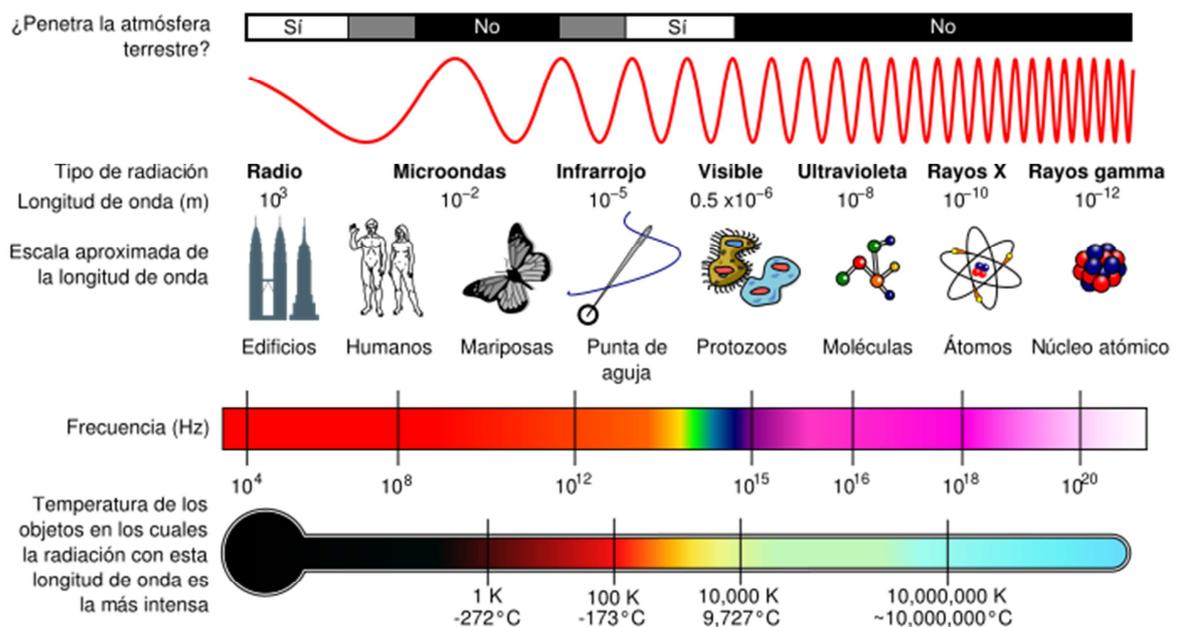


Figura. 1.11. Espectro Electromagnético

1.2.3.2 Conversión A/D

Convierte la señal obtenida del dispositivo de captura en una señal digital

1.2.3.3 Memoria de Video

Memoria de tipo RAM en la que se almacena la imagen digitalizada. Normalmente la memoria A/D y la memoria de video se agrupan en un solo módulo conocido como *Frame Grabber* (captor de imágenes)

1.2.3.4 Procesador

La memoria de video se acopla a un procesador que permite operar sobre la imagen. Opcionalmente puede existir otro procesador dedicado a la captura y procesamiento de imágenes.

1.2.3.5 Monitor

Generalmente se tiene un monitor que permite visualizar las imágenes adquiridas. El procesador y el monitor pueden ser parte de una computadora de propósito general a la que se ha acoplado el captor de imágenes.

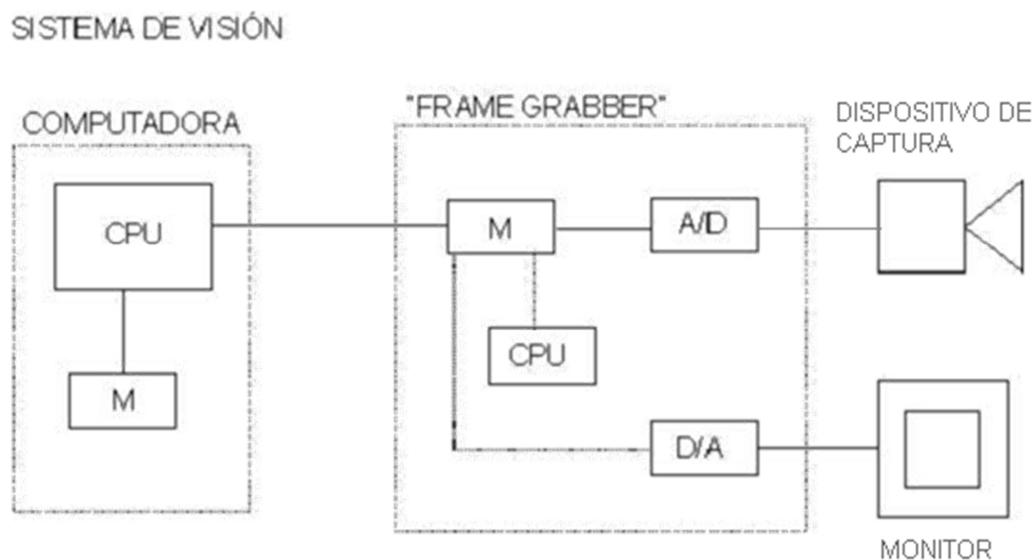


Figura. 1.12. Arquitectura Básica Sistema de Visión

1.2.4 Niveles de análisis

David Marr, es uno de los precursores de la visión computacional y uno de los mayores investigadores de esta ciencia. Para él al considerar la visión computacional existen tres niveles de análisis. Estos son:

- **Teoría Computacional.** (¿El qué?) El objetivo del proceso computacional, sus metas y las estrategias adecuadas para realizarlo.
- **Representación y Algoritmo.** (El ¿cómo? conceptual) La descripción del proceso computacional, el representar las entradas y salidas, proponer el algoritmo para dicha transformación
- **Implementación.** (El ¿cómo? físico) Como se realiza físicamente dicho proceso.

Analizar un proceso a los diferentes niveles ayuda a un mejor entendimiento y realización.

1.2.5 Niveles de visión

Como ya se dijo visión computacional es que a partir de una imagen se llegue a una descripción de la misma en base a un interés particular. El proceso para realizar esta descripción se lo puede dividir en varios niveles, en cada una se va refinando y reduciendo la cantidad de información hasta llegar a la descripción deseada. Generalmente se consideran tres niveles:

- **Procesamiento de bajo nivel.** Se trabaja directamente sobre cada pixel para extraer propiedades como orillas, gradiente, profundidad, textura, color, etc.
- **Procesamiento de nivel intermedio.** Consiste en agrupar los pixeles de acuerdo a una característica de bajo nivel para obtener líneas o regiones con el propósito de segmentarlas de la imagen de origen.
- **Procesamiento de alto nivel.** Está orientado al proceso de interpretación de los entes obtenidos en los niveles inferiores y se utilizan modelos y/o conocimientos previos del dominio.

1.3 VISIÓN ESTEREOSCÓPICA

1.3.1 Definición y Principios

La visión estereoscópica es una técnica en la que se utilizan dos imágenes, en analogía con el sistema de visión biológico, para recuperar información acerca de la profundidad de una escena. El algoritmo básico de esta técnica se describe a continuación:

- a) Obtener dos imágenes de la escena separadas por una distancia d .
- b) Identificar puntos correspondientes.
- c) Medición de la disparidad.
- d) Obtención de la profundidad en base a la geometría del sistema estereoscópico.

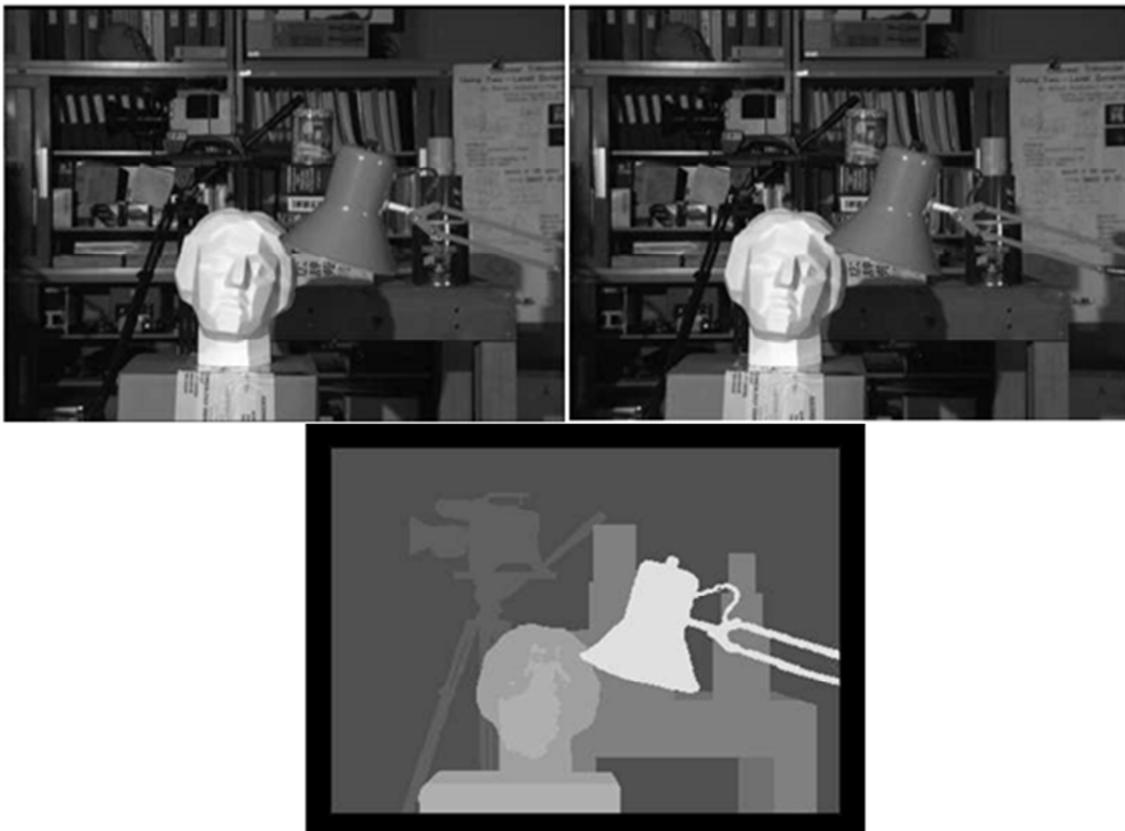


Figura. 1.13. Ejemplo algoritmo estéreo básico

Aunque pueda parecer que cumplir este algoritmo sea sencillo, el mayor problema de la visión estereoscópica lo constituye el paso b, ya que en él se debe identificar que pixel de la una imagen corresponde a la otra imagen para obtener su posición. La identificación de esta correspondencia es el principal objetivo de los algoritmos de visión estéreo, esto se resume en la ecuación 1.1.

$$\text{Pixel}_{L(x\#,y\#)} \xrightarrow{\text{corresponde}} \text{Pixel}_{R(x?,y?)} \quad (1.1)$$

Los otros pasos (a, c, d) se refieren a la geometría del sistema estereoscópico, pudiendo simplificar su cumplimiento en base a la posición de las cámaras.

1.3.2 Geometría del sistema estéreo y principio de obtención de la distancia

La obtención de la distancia en un sistema estereoscópico está íntimamente ligada a la geometría del mismo, por lo tanto se describirá a la vez tanto la geometría del sistema estéreo como su principio de funcionamiento. La geometría de un sistema estereoscópico puede diseñarse de forma que las cámaras tengan sus ejes ópticos paralelos (los ejes ópticos nunca se llegan a intersectar) o convergentes (ejes ópticos llegan a intersectarse).

En un sistema que tiene los ejes ópticos paralelos la percepción de distancia para objetos que se sitúan en la lejanía es mejor que la noción de distancia para objetos de interés cercanos. Por el contrario, en un sistema que tiene los ejes ópticos convergentes, la percepción de distancia para objetos que se sitúan en la cercanía es mejor que la noción de distancia para objetos de interés lejanos.

El modelo geométrico más utilizado en visión artificial es el de ejes ópticos paralelos, sobre él se han desarrollado la mayoría de teorías y algoritmos de estos sistemas de visión

1.3.2.1 Geometría de ejes paralelos

Como ya se menciona la geometría estéreo depende de la forma en que las cámaras son ubicadas, el caso más generalmente aplicado, más estudiado y sencillo es aquel en el que se cumplen las siguientes condiciones:

- Los ejes de profundidad (ejes ópticos) de las cámaras son paralelos.
- Las cámaras se encuentran separadas una distancia b en una línea paralela al plano formado entre los ejes paralelos (Z_I, Z_D), es decir solo existe desplazamiento en uno de los ejes entre las dos cámaras, se puede apreciar en la figura 1.14.

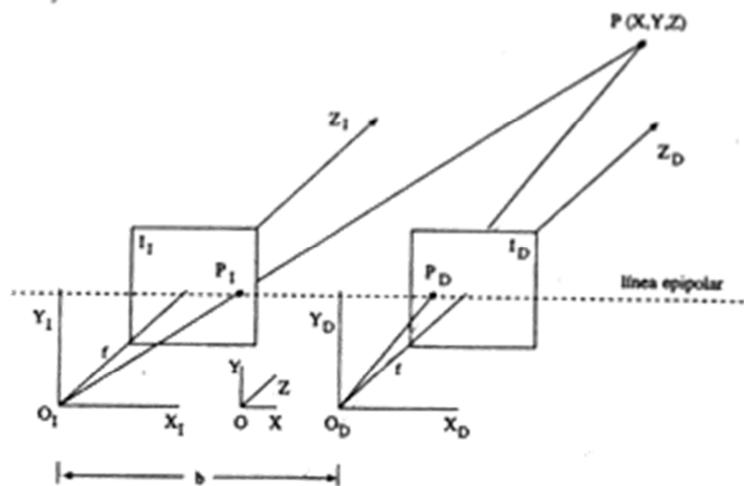


Figura. 1.14. Geometría de ejes paralelos

La figura 1.15 muestra una visión desde arriba de un sistema estéreo compuesto por dos cámaras. Los planos imagen, izquierdo y derecho, son coplanares y están representados por los segmentos I_I y I_D respectivamente. O_I y O_D son los centros de proyección.

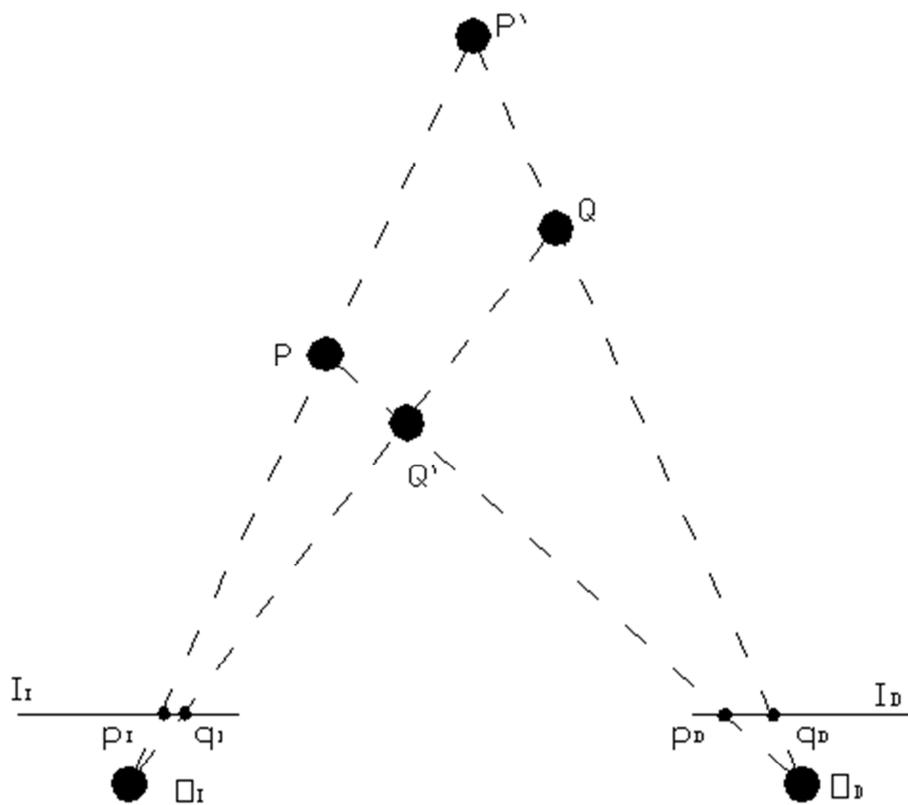


Figura. 1.15. Triangulación y Correspondencia

La forma en la cual la visión estereoscópica determina la posición en el espacio de P y Q , en la figura 1.15, es por triangulación es decir; por intersección de los rayos definidos por los centros de proyección de cada una de las cámaras y los correspondientes puntos de proyección en las retinas de P y Q , p_I , p_D , q_I , q_D respectivamente. La triangulación depende de forma muy importante del problema de las correspondencias: Si p_I corresponde a p_D y q_I corresponde a q_D la intersección de los rayos proyectados $O_I p_I$ con $O_D p_D$ y $O_I q_I$ con $O_D q_D$ conduce a interpretar los puntos P y Q en la posición correcta; pero si p_I corresponde a q_D y q_I a p_D , la triangulación nos devuelve los puntos P' y Q' .

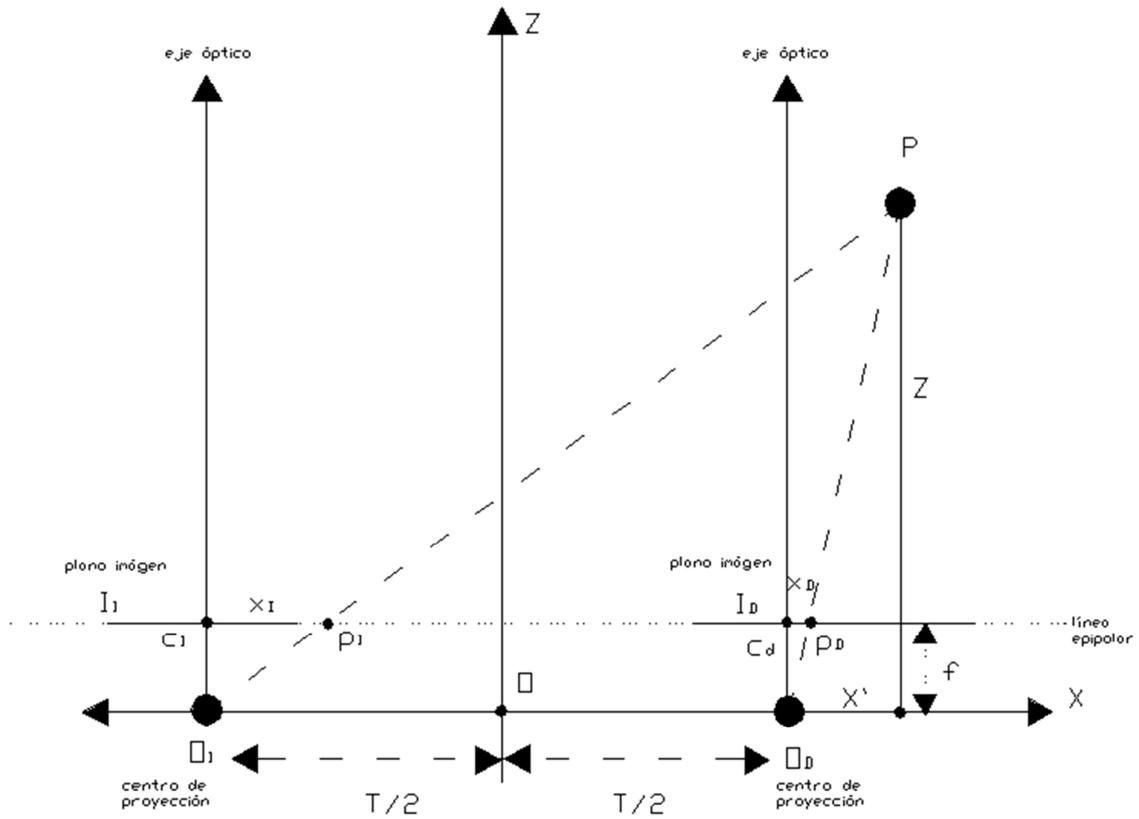


Figura. 1.16. Cálculo de la profundidad

En la figura 1.16, se explica el análisis geométrico de la determinación de la profundidad. Identifiquemos por T a la distancia entre los centros de proyección. Sean x_1 y x_2 las coordenadas de p_1 y p_2 respecto a los puntos principales c_1 y c_2 , f la distancia focal común y X' a la proyección de la distancia entre el origen y el punto P en el eje X . De las relaciones entre triángulos similares (p_1, c_1, O_1) con (O_1, P, X') y (p_2, c_2, O_2) con (O_2, P, X') tenemos:

$$\begin{aligned}
\frac{x_I}{X' + \frac{T}{2}} &= \frac{f}{Z} & \frac{x_D}{X' - \frac{T}{2}} &= \frac{f}{Z} \\
\frac{x_I}{f} &= \frac{X' + \frac{T}{2}}{Z} & \frac{x_D}{f} &= \frac{X' - \frac{T}{2}}{Z} \\
x_I &= \frac{f}{Z} \left(X' + \frac{T}{2} \right) & x_D &= \frac{f}{Z} \left(X' - \frac{T}{2} \right) \\
d &= x_I - x_D \\
d &= \frac{f}{Z} \left(X' + \frac{T}{2} - X' + \frac{T}{2} \right) \\
d &= \frac{f}{Z} (T) \\
Z &= \frac{f * T}{d} \\
Z &= f \frac{T}{d}
\end{aligned} \tag{1.2}$$

Donde d es la disparidad, de la ecuación anterior vemos que: “La profundidad es inversamente proporcional a la disparidad”

1.3.3 Algoritmos de visión estéreo o el problema de la correspondencia

Para encontrar la correspondencia entre los píxeles de las dos imágenes se utilizan diferentes métodos o algoritmos que vamos a enunciar y describir superficialmente, sin embargo existen restricciones o hipótesis que son generalmente aplicados en los algoritmos para encontrar dicha correspondencia. Estas restricciones están basadas en propiedades físicas y geométricas razonables de las escenas.

1.3.3.1 Restricciones Aplicadas al problema de correspondencia

- **Restricción Epipolar.** Viene dada por la geometría epipolar del sistema estéreo, implica que el punto correspondiente de una imagen debe estar en la línea epipolar del punto en la otra imagen. Esta restricción reduce la búsqueda de toda la imagen a esta línea. Ver figura 1.14.

- **Restricción de Orden.** Supone el hecho de que la escena original está compuesta por objetos opacos y suficientemente grandes para que las proyecciones conserven el orden de apareamiento en cada imagen, como se puede ver en la figura 1.17(a) se cumple la restricción de orden mientras en la figura 1.17(b) se incumple.

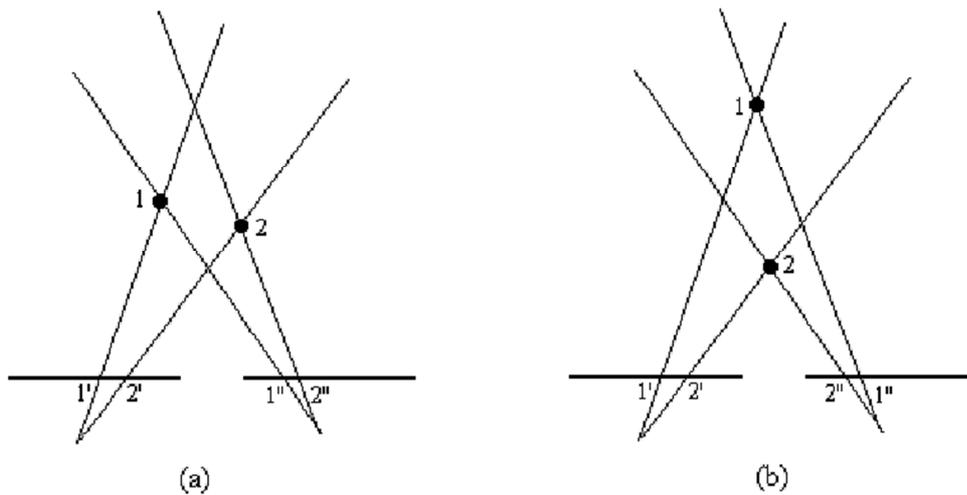


Figura. 1.17. Restricción de orden

- **Restricción de Unicidad.** Implica que un punto de la imagen tiene únicamente un punto correspondiente en la otra imagen, de no existir este punto correspondiente significa que éste está oculto en la otra imagen.

1.3.3.2 Clasificación de algoritmos de correspondencia

Según Brown [8] y otros, los algoritmos de visión estereó se pueden clasificar según como es utilizada la información para encontrar los puntos correspondientes en las líneas de búsqueda. La clasificación mas general es en aquellos que utilizan restricciones locales en una ventana alrededor del punto en cuestión y aquellos que imponen restricciones globales en la línea de búsqueda o la imagen.

Algoritmos Locales. Son eficientes computacionalmente, aunque en comparación con los métodos globales presentan dificultades para trabajar con imágenes que tengan regiones localmente ambiguas por ejemplo: oclusiones y

regiones de textura uniforme. Los algoritmos locales se clasifican según el método que utilizan para la detección de los puntos correspondientes en:

- **Block matching.** Se basa en encontrar el punto correspondiente comparando una región alrededor del punto con un conjunto de regiones iguales en la otra imagen. La región que presenta mayor semejanza es elegida y selecciona el punto correspondiente. Las medidas de semejanza usadas generalmente son: Correlación Normalizada (NCC), Suma de diferencias al cuadrado (SSD), Suma de valores absolutos (SAD), el costo computacional de cada una de estas medidas de semejanza se puede ver en la figura 1.18.

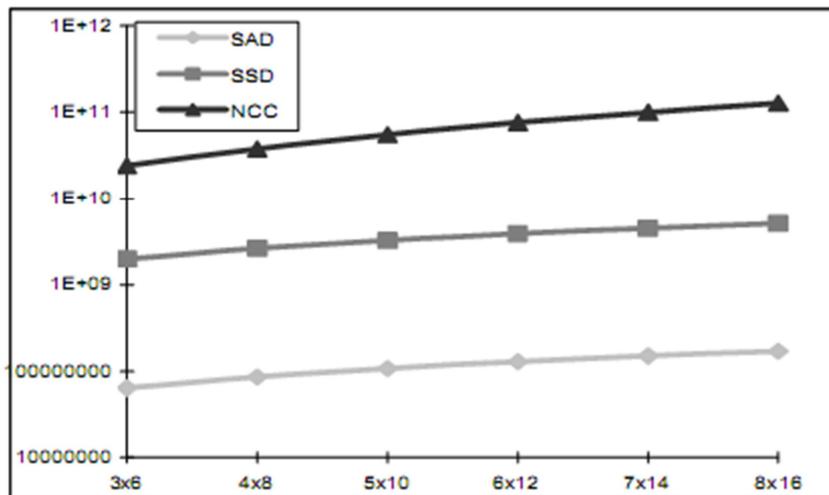


Figura. 1.18. Costo Computacional

- **Flujo Óptico.** Se basa en plantear una ecuación diferencial que relaciona el desplazamiento, d , de un pixel entre las imágenes izquierda y derecha, con el movimiento, asumiendo que su intensidad no varía.

$$\nabla_x I d + I_t = 0 \quad (1.3)$$

Donde $\nabla_x I$ es el gradiente horizontal de la imagen e I_t es la derivada temporal. El desplazamiento vertical del pixel se asume que es nulo dada la configuración del par estéreo.

- **Correspondencia de Características.** Se basa en buscar los puntos correspondientes en regiones de la imagen donde existen características relevantes (vértices, bordes, figuras, etc.) Estos puntos no son muchos, y los mapas de disparidad que se pueden calcular no son densos; se obtienen puntos en el espacio imagen pero no un mapa de disparidad con la profundidad estimada de cada punto de la imagen

La complejidad de los métodos de correspondencia de características y flujo óptico es comparable y mayor al de correspondencia de características, su principal ventaja en comparación a este, es poder calcular mapas de disparidad densos.

Algoritmos Globales. Imponen restricciones globales en la minimización de alguna expresión de costo o energía que modele el fenómeno estéreo, reduciendo los errores en las regiones con problemas. Estos modelos utilizan comúnmente dos tipos de búsqueda:

- **Programación Dinámica (DP).** En este método se reduce la complejidad de cálculo en problemas de optimización descomponiendo el problema en sub problemas menores. Las restricciones globales que se imponen con este método son, generalmente, la restricción epipolar y la restricción de orden. Para esto se construye una representación de las posibles correspondencias para cada punto construyendo una imagen, que se denomina, imagen del espacio de disparidad, donde se busca un camino que recorra este espacio y minimice un cierto costo.
- **Corte de Grafos (GC).** Se basa en armar un grafo a partir de los datos de las imágenes y buscar un corte mínimo. Dependiendo de cómo se arma el grafo, el resultado obtenido es la minimización de una cierta expresión de energía. Estos métodos requieren un costo computacional mayor de la programación dinámica.

Según Federico Lecumberry [8], ninguno de los algoritmos globales es óptimo para cualquier tipo de escenas y pueden presentar errores en algunas partes del mapa de disparidad. Como sucede con muchos algoritmos, el funcionamiento es óptimo con cierta estructura geométrica de la escena, pero cuando alguna de las restricciones en la estructura o hipótesis de la escena no se verifica, se generan errores. El tiempo de ejecución de DP es varias decenas de veces menor que GC. En las pruebas que se realizaron, los tiempos promedio fueron de 450 ms. contra 29.2 s., promedio en unas 140 pruebas con cada algoritmo.

1.3.4 Consideraciones de sistemas de visión estéreo

1.3.4.1 Tiempo Real en Visión Estéreo

Los algoritmos para el cálculo de mapas de disparidad densos en tiempo real generalmente están asociados a algún tipo de hardware específico (DSP, FPGA) o microprocesadores que permita realizar el procesamiento de datos en forma eficiente. Los métodos que implementan, son algoritmos de los clasificados por imponer restricciones locales basados generalmente en encontrar correspondientes utilizando alguna medida de semejanza (SSD, SAD) en un proceso de *block matching*, y agregando consistencia entre los correspondientes obtenidos en las dos imágenes.

1.3.4.2 Velocidad o FPS de un sistema estéreo

La velocidad de un sistema de visión se mide en FPS (*Frames per second*) que representa la cantidad de imágenes procesadas por segundo por el sistema. Sin tomar en cuenta al algoritmo utilizado esta velocidad disminuye por dos factores fundamentales del sistema que son:

- **Número de disparidades.** Es la cantidad de niveles de profundidad que se representan.
- **Resolución.** Es la cantidad de pixeles disponibles para representar las imágenes.

En la tabla 1.1 se puede apreciar, la disminución de la velocidad de un sistema de visión comercial en base a estas dos características.

Resolución	Número de Disparidades	FPS PII 233MHz	FPS PIII 500 MHz
160x120	16	90	
160x120	32	45	
320x240	16	23	40
320x240	32	12	22
640x480	32		4.8
640x480	48		

Tabla. 1.1. FPS sistema comercial

1.4 SISTEMA ROBÓTICO INDUSTRIAL CRS A-255

1.4.1 Brazo robótico articulado

El sistema robótico CRS de la serie A255 consta de un brazo robótico articulado TRA-14-203, presenta una configuración antropomórfica y consiste de 5 articulaciones rotacionales, por lo tanto 5 grados de libertad. Como se puede apreciar en la figura 1.19.

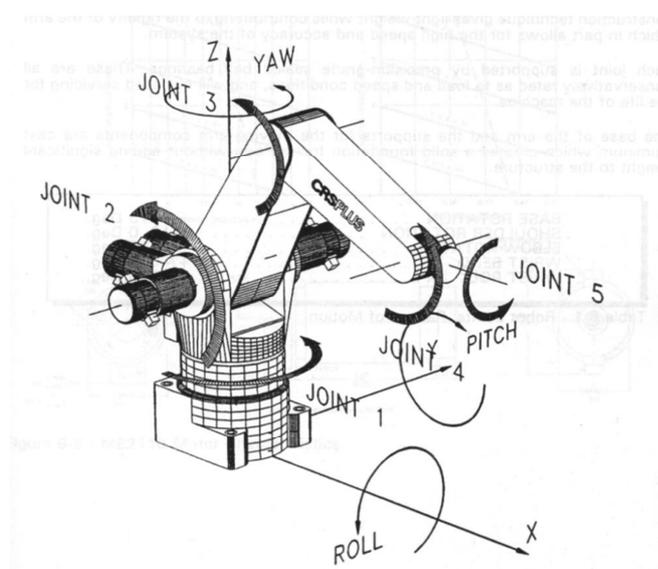


Figura. 1.19. Brazo articulado TRA-14-203

Las posiciones y orientaciones que se pueden alcanzar con el robot, es decir su volumen de trabajo, siempre se refieren a la terminación del robot o punto TCP, "*Tool Centre Point*", o punto central de herramienta, generalmente en este punto se encuentra la herramienta correspondiente con la que el robot puede realizar acciones con el medio. El TCP puede ser referido a varios sistemas coordenados y a diferentes orígenes de referencia. A continuación se ilustra los diferentes sistemas coordenados, referidos al sistema de coordenadas por defecto y los comandos que permiten modificar la posición y orientación de estos sistemas.

1.4.1.1 Sistema Coordenado del Mundo

Este sistema de coordenadas de posición consta de tres ejes cartesianos ortogonales llamados X, Y y Z, el origen de coordenadas se ubica en el centro de rotación de la primera articulación sobre la superficie de montaje del robot, las direcciones positivas de los ejes cartesianos están definidas por la regla de la mano derecha.

Por otro lado la orientación del último eslabón o muñeca está definida por tres ángulos, la dirección de aumento de estos ángulos también se define por la regla de la mano derecha:

- **Yaw.** Indica el ángulo de orientación respecto al eje Z
- **Pitch.** Indica el ángulo de orientación respecto al eje Y
- **Roll.** Indica el ángulo de orientación respecto al eje X

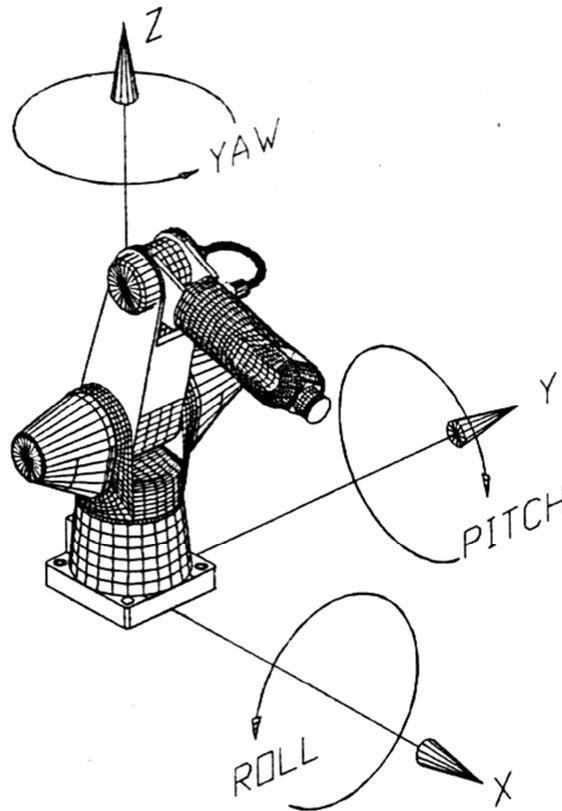


Figura. 1.20. Sistema Coordinado del Mundo

La posición del origen de coordenadas y la orientación del sistema pueden ser alteradas utilizando el comando OFFSET.

Los comandos de movimiento que hacen referencia a este sistema de coordenadas son: JOG, MOVE, X, Y, Z, PITCH, ROLL, YAW.

1.4.1.2 Sistema Coordinado de la Herramienta

Este sistema de coordenadas consta de tres ejes cartesianos ortogonales llamados X, Y, Z, el origen de coordenadas es TCP se ubica en el centro de la superficie externa de la brida de la herramienta. El eje X es normal a la brida de la herramienta en el origen mientras que los ejes Y y Z son ortogonales a X y están alineados a los ejes Y y Z del sistema de coordenadas del mundo. La orientación del último eslabón o muñeca está definida dada por los ángulos, Yaw, Pitch y Roll al igual que en el sistema de coordenadas del mundo.

Mediante el comando TOOL se puede definir un nuevo TCP, especificando la posición y orientación relativa al sistema de coordenadas de la herramienta por defecto. Este sistema de coordenadas es móvil, es decir, se mueve junto con la brida de la herramienta o junto al TCP especificado, durante la traslación del manipulador, a diferencia del sistema de coordenadas del mundo que es fijo.

Los comandos de movimiento que hacen referencia a este sistema de coordenadas son: APPRO, DEPART.

1.4.1.3 Sistema Coordenado Cilíndrico

El sistema de coordenadas cilíndrico usa los mismos ejes Z, Yaw, Pitch y Roll que el sistema de coordenadas del mundo. Sin embargo en lugar de los ejes X y Y este sistema de coordenadas utiliza las coordenadas polares θ (theta) y R (radio). Theta es el ángulo de posición alrededor del eje Z, mientras R es la distancia en la dirección dada por theta, ver figura 1.21.

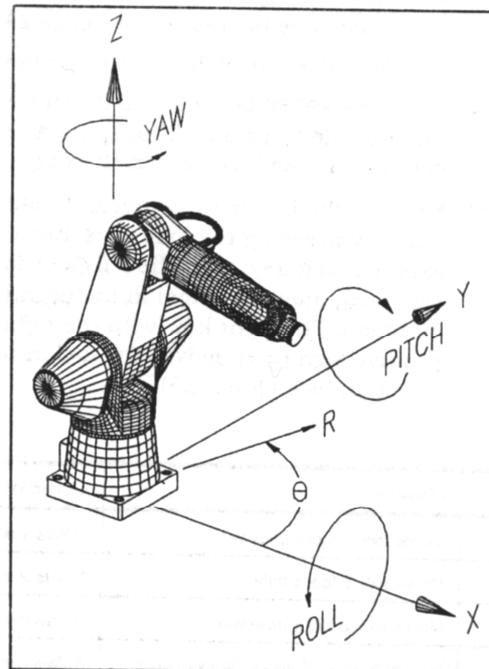


Figura. 1.21. Sistema Coordenado Cilíndrico

Este sistema de coordenadas no tiene ningún comando de movimiento a través de instrucciones de RAPL-II, únicamente puede moverse en los mencionados ejes a través del modo manual del *teach pendant*.

1.4.1.4 Sistema de Coordenadas de las articulaciones

Al ser todas las articulaciones del manipulador de tipo rotacional se puede establecer a los diferentes ángulos del manipulador como un sistema coordinado. En este sistema coordinado la posición angular de cada una de las articulaciones define la posición del robot. El sentido de giro y el nombre de cada una de las articulaciones se puede observar en el gráfico:

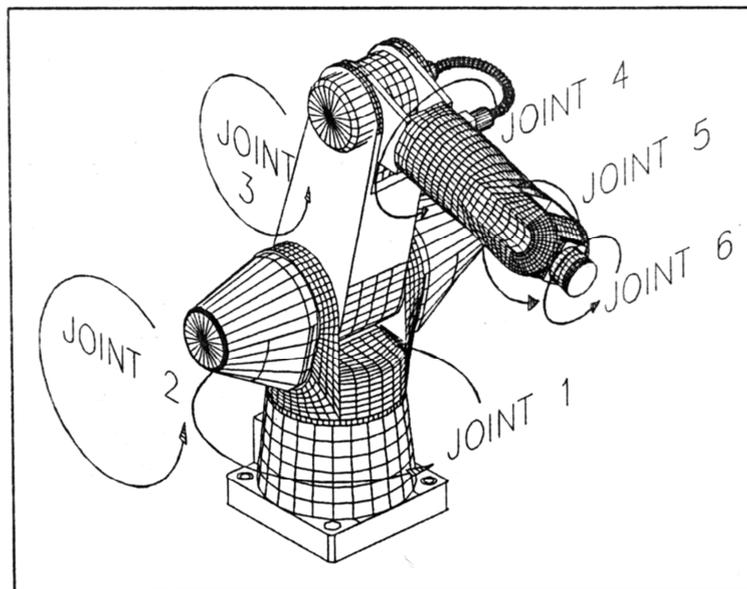


Figura. 1.22. Sistema Coordinado de las articulaciones

Los comandos de movimiento que hacen referencia a este sistema de coordenadas son: JOINT, MA, MI.

1.4.2 Controlador robótico C500

En la figura 1.23 se puede visualizar el Panel frontal del controlador C500 de la compañía CRS.

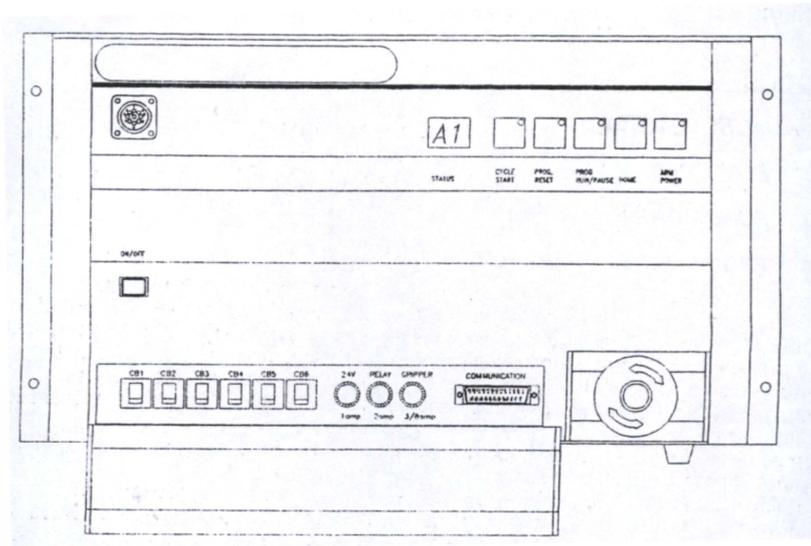


Figura. 1.23. Panel Frontal Controlador C500

1.4.2.1 Interfaz de operación

El controlador C500 opera con las siguientes interfaces al usuario:

- El panel frontal del controlador
- El teach pendant (terminal de aprendizaje)
- El canal interactivo de comunicación

El panel frontal del controlador no es una interfaz de programación, por lo tanto no se lo describirá. El terminal de aprendizaje permite ejecutar todos los comandos del lenguaje RAPL-II y programar el controlador C500. El canal interactivo de programación permita tanto ejecutar comandos y la programación del controlador C500, requiere de un computador que ejecute el programa ROBCOMM el cual, a través del puerto serie, ejecutará comandos del lenguaje RAPLII o permitirá la creación de programas en base a estos comandos. Existen tres niveles de control, para este tipo de interfaz, estos son:

- **Inmediato (I).** Usado para la ejecución de comandos individuales. Cuando ningún programa está corriendo, y no se ha seleccionado el modo manual, entonces el controlador está en modo inmediato. Permite el modo de edición
- **Manual (M).** Usado para la manipulación manual del manipulador. Este modo permite mover al manipulador tanto desde el terminal de aprendizaje así como desde el teclado. Muestra diferentes mensajes de línea de comandos (*prompt*) para indicar en qué modo se encuentra el controlador. Existen cuatro modos en los que se puede operar: *Joint Manual Mode*, *World Manual Mode*, *Tool Manual Mode* y *Cylindrical Manual Mode*. Permite el modo de edición. Los mensajes de línea de comando son:

>* Modo de edición
>> Modo inmediato
J> Joint Manual Mode
C> Cylindrical Manual Mode
W> World Cartesian Manual Mode
T> Tool Cartesian Manual Mode
O> ONLINE motion command mode

- **Programa (P).** El modo de programa permite al usuario ejecutar programas almacenados en la memoria de usuario del controlador.

1.4.2.2 Comunicación

El controlador C500 posee tres medios de comunicación que son:

- **Puerto de propósito general (GPIO).** Son 16 entradas y salidas que pueden ser utilizadas para la interface del controlador con otras maquinas, que podrían ser controladas por el C500 o solamente monitoreadas.
- **Sistema I/O, o conector (SYSIO).** Contiene señales que imitan los pulsadores del panel frontal e indicadores led. Adicionalmente, el conector SYSIO contiene un par de contactos de parada de emergencia que

permite instalar de forma remota un sistema de apagado de energía de las articulaciones del C500 en casos especiales. En la figura 1.24 se aprecia los dos puertos antes mencionados y su ubicación.

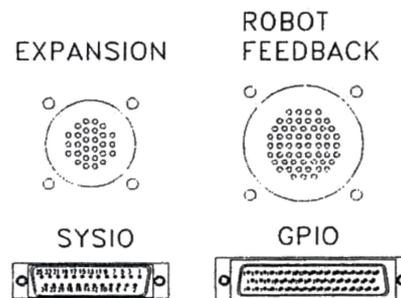


Figura. 1.24. Vista del Panel posterior del Controlador C500.

- **Puerto de Comunicaciones.** El controlador C500 puede conectarse a cualquier PC mediante el puerto serial a través de una interfaz llamada ACI (*Advanced Communication Interface*)

ACI es un protocolo de comunicación serial utilizado en el controlador C500, el cual fue desarrollado por CRS. Utiliza un método de transferencia especial que es diferente de la comunicación serial tradicional.

La comunicación ACI es un bloque estructurado RS-232/422/485 basado en un protocolo que permite cargar datos de la memoria del robot al computador o viceversa. ACI permite comunicar sistemas computarizados externos con uno o más sistemas robóticos por medio de una señal serial (RS232 o RS422).

Este protocolo se utiliza para transferir datos hacia o desde el dispositivo máster. Permite el direccionamiento a segmentos de la memoria 8086, la cual va a ser el medio para la comunicación. Realiza la comprobación de errores y reintentos de transmisión automáticamente con el fin de garantizar un enlace de comunicación confiable.

ACI es un protocolo máster/esclavo. Todos los controladores son configurados como esclavos en una red. Algunos computadores externos podrían ser configurados como máster. Solo un máster puede inicializar la comunicación.

Cuando el controlador C500 enciende, el estado ACI es por defecto en el puerto de comunicaciones del panel frontal. Si en ROBCOMM-II se ingresa por medio de modo terminal, la función de este puerto cambia a estado interactivo. Al salir del modo terminal se restaurara el estado de protocolo ACI en el puerto de comunicaciones. La comunicación consiste de cuatro bloques separados, que serán descritos a continuación:

- El dispositivo máster establece un enlace de comunicación para especificar un número de dispositivo esclavo como un objetivo. El dispositivo esclavo confirma esta existencia en el canal serial, y el dispositivo máster confirma al objetivo que se estableció la comunicación.
- Después de establecer el enlace, el máster proporciona al esclavo información que describe la transferencia de datos que está a punto de llevarse a cabo.
- Luego de la recepción de las especificaciones para la transferencia de datos, el dispositivo esclavo manda una confirmación y el dato actual es enviado. El dato es transferido en paquetes (bloques) de 128 bytes. Cada bloque de datos contiene su carácter de comienzo y parada además de un byte de validación para después de la transferencia.
- Después de que el dato ah sido recibido satisfactoriamente, el dispositivo máster cierra la comunicación con un carácter EOT.

CAPÍTULO 2

SISTEMA DE VISIÓN ESTEREOSCÓPICA

2.1 INTRODUCCIÓN

El presente capítulo describirá el diseño e implementación del sistema de visión estéreo. Por ello a modo introductorio se definirán las variables que deben obtenerse, las características primordiales con las que deberán contar estas variables, los requerimientos de operación del sistema, así como las limitantes inherentes al proyecto. En base a los requisitos de estas secciones expuestas, se definirá el diseño del sistema para luego desarrollar la implementación del mismo.

2.1.1 Variables a obtenerse y características

El objetivo fundamental de un sistema de visión estéreo es extraer de la escena la característica de profundidad de las entidades que la componen. Es así que para que el brazo robótico realice la tarea de seguimiento de un objeto, necesitamos conocer la posición espacial de este objeto. Ya que el sistema de adquisición de imágenes se ubicará como una herramienta del manipulador robótico, el sistema de coordenadas de la herramienta será elegido como referencia de la posición espacial de este objeto. En consecuencia los resultados fundamentales que se deben obtener del sistema de visión son:

Distancia en X. Es la distancia de separación en profundidad entre el centro de las cámaras y el objeto a seguir, este es el resultado principal del sistema de visión, por tratarse de una tarea de seguimiento el sistema de visión deberá garantizar las siguientes características mínimas de resultado:

- Rango Nominal: 20 cm a 100 cm.
- Resolución: 1cm.
- Transparencia: alta
- Tiempo de respuesta: 100 ms
- Exactitud: alta, principalmente en mediciones próximas (20 a 30 cm)

Distancia en Y. Distancia de separación horizontal del objeto con el centro de las cámaras.

- Rango Nominal: 10 cm
- Resolución: 1 cm
- Transparencia: alta
- Tiempo de respuesta: 100 ms
- Exactitud: alta, principalmente en mediciones próximas (20 a 30 cm)

Distancia en Z. Distancia de separación vertical del objeto con el centro de las cámaras, comparte las características del anterior resultado. Ver figura 2.1.

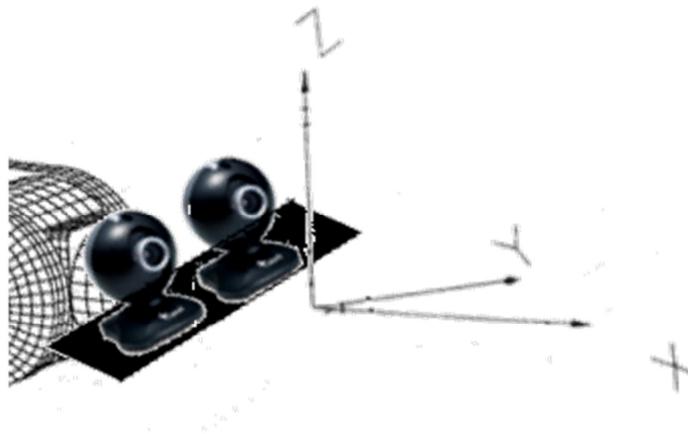


Figura. 2.1. Ejes de ubicación de objeto

2.1.2 Requerimientos de operación del sistema

Los requerimientos de operación del sistema se refieren a las condiciones de trabajo en las que se va a desenvolver el sistema de visión así como las características que un usuario podría demandar para su utilización.

2.1.2.1 Condiciones de trabajo

El sistema de visión trabajara en conjunto con uno de los brazos robóticos CRS-A255 del laboratorio de robótica del departamento de electrónica, por lo tanto el ambiente de trabajo tiene las siguientes características:

- Iluminación promedio en el área de trabajo: 200 lux.
- Temperatura nominal: 23 grados centígrados
- Residuos ambientales: Polvo fino
- Ambiente Inflamable: No
- Utilización en la intemperie: No
- Plataforma móvil – tipo de movimiento: Si – moderado

2.1.2.2 Condiciones de Usuario

Al ser un sistema de tipo pedagógico se espera cumplir las siguientes demandas:

- Visualización de la escena desde la aplicación
- Configuración de parámetros de la tarea
- Versatilidad para seguir diferentes tipos de objeto
- Monitorización de parámetros de interés

2.1.3 Limitantes inherentes al proyecto

En primer lugar se debe mencionar que el proyecto es de tipo investigativo, con el se pretende proporcionar una herramienta educativa en el campo de visión estereoscópica. Al hablar de sistemas de visión estereoscópica comerciales se debe entender que estos representan soluciones específicas del problema, si bien es cierto, pueden tener capacidades de versatilidad asociadas, son siempre sistema de arquitectura cerrada; la modificación física de los elementos de adquisición así como la interpretación del software para una posterior modificación son tareas difíciles de realizar y que pueden comprometer el desempeño del sistema.

Por otro lado el costo de estos sistemas de visión es elevado, a partir de \$1000 dólares en su país, ver anexo A, de origen, además se estaría comprando tanto los elementos de adquisición, como el software procesador de la escena. De realizarse la compra de uno de estos equipos se estaría desperdiciando el software procesador ya que el afán del proyecto es desarrollar un software propio y que se ajuste a los requisitos de la tarea a realizarse y el entorno de aplicación.

Como ya se mencionó el sistema de visión tiene una función educativa, y no de producción. Es decir estará expuesto a la experimentación por parte de los alumnos, esto implica un riesgo para cualquier sistema. Este riesgo aumenta si los equipos fueran costosos y sin representación dentro del país

En este sentido, son los elementos de adquisición de imágenes, cámaras, los que deben analizarse. Las características de estos elementos pueden perjudicar el rendimiento de todo el sistema. Con esto no se quiere decir que las cámaras necesariamente tienen que ser costosas, sino que deben evaluarse sus características y desempeño respecto al algoritmo de extracción de las características de interés.

2.2 SELECCIÓN DEL ALGORITMO DE CORRESPONDENCIA

Al ser la correspondencia el principal problema que tiene que solucionarse en un proceso de visión estereoscópica, los investigadores han desarrollado diferentes algoritmos para solucionarlo. Estos algoritmos tienen diferentes características y requisitos de utilización, en esta sección se analizará cuál de ellos es el que más conviene a este proyecto en particular. Los parámetros más importantes que serán analizados son:

- Costo Computacional
- Versatilidad
- Relación con la tarea del proyecto
- Requisitos de utilización y rentabilidad

2.2.1 Costo Computacional

Según la información obtenida en el capítulo uno, se puede resumir los siguientes puntos acerca del costo computacional de los diferentes algoritmos de correspondencia:

- Los algoritmos locales son menos costosos computacionalmente que los algoritmos globales.
- Respecto a métodos globales el tiempo de ejecución de Programación Dinámica es varias decenas de veces menor que Corte de Grafos. En las pruebas que se realizaron, los tiempos promedio fueron de 450 ms. contra 29.2 s. promedio
- El método local de correspondencia de bloques utiliza tres medidas de semejanza, en orden creciente respecto al costo computacional son: SAD, SSD y NCC (ver figura 1.18).

Como se puede observar el tiempo de ejecución de los distintos algoritmos va directamente relacionado con su coste computacional, al tratarse de una tarea de seguimiento es de interés para el proyecto seleccionar el algoritmo de menor coste computacional. Los algoritmos globales tienen tiempos de ejecución superiores a los 100ms que se estiman como parámetro necesario en el conocimiento de la profundidad.

2.2.2 Versatilidad

Con base en la información del capítulo uno, se puede resumir los siguientes puntos acerca de la versatilidad de los diferentes algoritmos de correspondencia:

- A diferencia de los algoritmos globales los algoritmos locales presentan dificultades para trabajar con imágenes que tengan regiones localmente ambiguas por ejemplo: oclusiones¹ y regiones de textura uniforme

¹ Oclusión.- Segmento de información gráfica oculto en la escena.

- Ninguno de los algoritmos globales es óptimo para cualquier tipo de escenas y pueden presentar errores en algunas partes del mapa de disparidad. Como sucede con muchos algoritmos, el funcionamiento es óptimo con cierta estructura geométrica de la escena

Se puede observar que los algoritmos globales son los más versátiles en comparación a los algoritmos locales, sin embargo también se define que ningún algoritmo es óptimo para todo tipo de escena.

2.2.3 Relación con la tarea del proyecto

La principal tarea a realizarse en este proyecto es la de seguimiento de un objeto, es decir el principal o único componente de la escena del que se necesita información respecto a profundidad es del objeto a seguir.

Por lo general todos los algoritmos de correspondencia crean mapas de disparidad densos. Estos mapas de disparidad pueden tener niveles muy limitados de disparidad, debido a que ha mayor cantidad de niveles de disparidad aumenta el tiempo de ejecución.

Los algoritmos que emplean *matching* de características por otro lado centran su atención en una característica de interés de la escena, por ejemplo el objeto, y además al analizar únicamente este punto pueden obtener resoluciones superiores de profundidad que los otros métodos.

2.2.4 Requisitos de utilización y rentabilidad

Como ya se mencionó el presupuesto del proyecto es limitado y estaría restringido a la adquisición de los elementos de adquisición de imágenes. Por lo tanto los requisitos de equipo del algoritmo escogido no deben comprometer la compra de equipos.

Rescatando puntos importantes de la descripción de los algoritmos de correspondencia podemos decir que los algoritmos para el cálculo de mapas de

disparidad densos en tiempo real generalmente están asociados a algún tipo de hardware específico (DSP, FPGA) o microprocesadores que permita realizar el procesamiento de datos en forma eficiente. Los métodos que implementan son algoritmos que imponen restricciones locales.

2.2.5 Algoritmo de correspondencia seleccionado

En base a los parámetros analizados se estima que el algoritmo más apropiado a nuestro proyecto es uno basado en correspondencia de características, ya que centra su atención en el objeto a seguir, obtiene niveles de disparidad de resoluciones muy altas y dependiendo de la característica del objeto a seguir su coste computacional puede ser muy bajo y por lo tanto no se requiere hardware especial de procesamiento. Su principal desventaja al ser un algoritmo de tipo local, es la versatilidad respecto a la escena, sin embargo al centrar la atención en características relevantes se podría superar esta desventaja incluyendo principalmente manejo de oclusiones respecto a la forma u objeto de interés.

2.3 ALGORITMO DE UBICACIÓN EN EL PLANO IMAGEN

Mediante el algoritmo de correspondencia se puede conocer la componente X de la posición espacial del objeto respecto a las cámaras, siendo este el principal objetivo de la visión estéreo. Ahora necesitamos encontrar las otras dos variables objetivo de nuestro sistema de visión es decir la distancia de separación horizontal, Y, y la distancia vertical, Z, del objeto con el centro de las cámaras.

Al detectar el objeto de interés en el algoritmo de correspondencia es posible encontrar las coordenadas del centro del objeto en pixeles, sin embargo, únicamente con este dato no es posible determinar la verdadera dimensión del objeto ó estimar la longitud real de las componentes de ubicación espacial, ver figura 2.2.

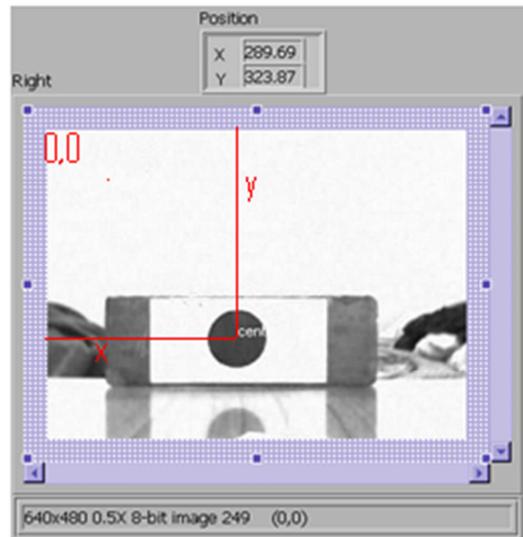


Figura. 2.2. Coordenadas en pixeles de un objeto

Esto se debe principalmente a que se desconoce la longitud real que representa cada pixel ya que depende de la distancia a la que se encuentren los objetos. Por ejemplo, si un objeto de radio R , se encuentra alejado del equipo de adquisición de imágenes, en el plano imagen se representa este radio R en menos cantidad de pixeles que si el objeto estuviera más próximo al equipo de adquisición de imágenes, ver figura 2.3.

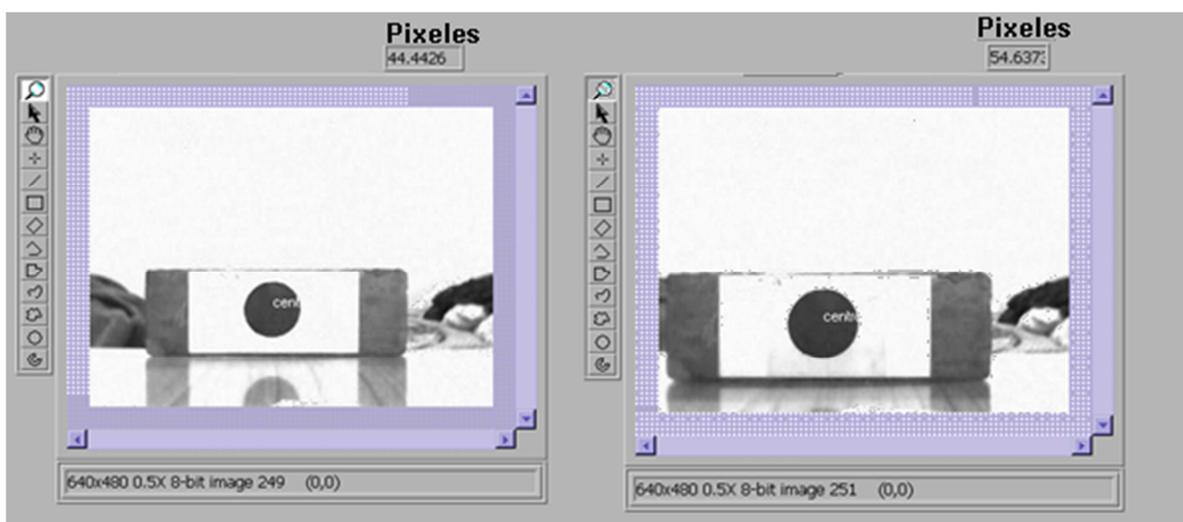


Figura. 2.3. Variación de pixeles

Es por ello que la principal tarea del algoritmo de ubicación en el plano imagen es conocer esta variable, de ahora en adelante llamada longitud – pixel.

2.3.1 Variable longitud – pixel

La variable longitud – pixel entonces podría definirse como la relación entre la longitud real y el número de pixeles que representa en una imagen digital, a cierta distancia.

$$VLP = \left[\frac{L}{pixel} \right] \quad (2.1)$$

Se proponen dos métodos para hallar esta relación, el primero se denominará de característica conocida y el segundo de aproximación por distancia, posteriormente se comparará estos dos métodos y se escogerá el más apropiado a nuestro proyecto.

2.3.2 Método de característica conocida

Como ya se mencionó, el principal objetivo de este método es conocer el valor de la variable longitud – pixel. A través del conocimiento estricto de las siguientes variables:

- Alguna característica de longitud real del objeto de interés
- La representación en pixeles de esta característica

Se puede obtener la variable longitud pixel dividiendo la longitud real entre la representación en pixeles de ella, por ejemplo:

Sea el objeto de interés una circunferencia de radio $R=3\text{cm}$, mediante procesamiento de imágenes se conoce que el radio de este objeto se representa en 40 pixeles a una distancia de 20 cm, y en 20 pixeles a una distancia de 40 cm. ¿Cuál es el valor de la variable longitud – pixel en las dos locaciones?

$$\text{A } 20 \text{ cm: } VLP = \frac{3\text{cm}}{40 \text{ pixel}} = 0.075 \frac{\text{cm}}{\text{pixel}}$$

$$\text{A } 40 \text{ cm: } VLP = \frac{3 \text{ cm}}{20 \text{ pixel}} = 0.15 \frac{\text{cm}}{\text{pixel}}$$

LONGITUD RADIO REAL 3 cm.

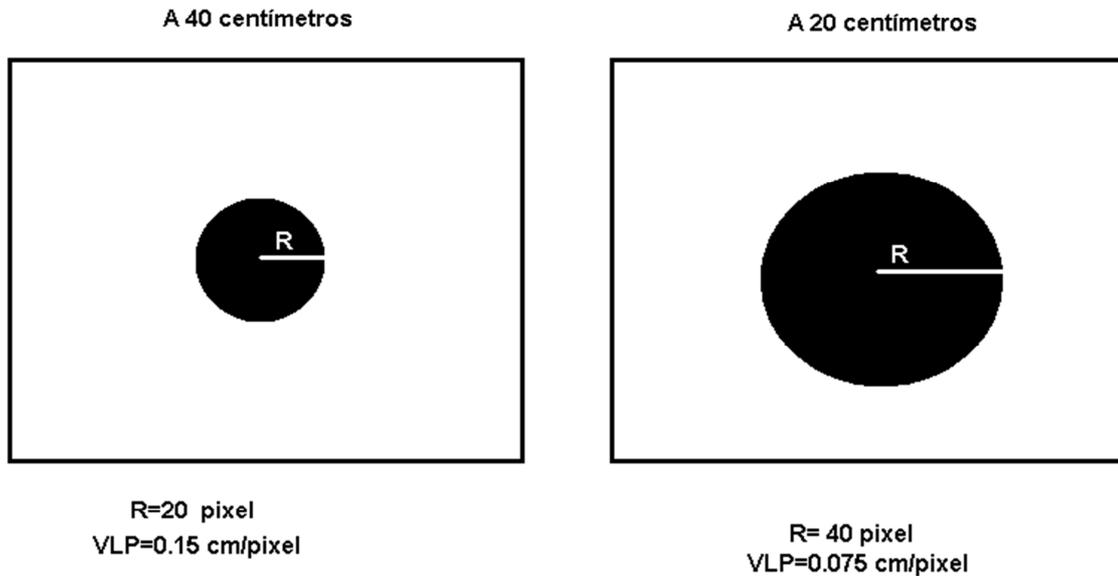


Figura. 2.4. Ejemplificación de la variación de VLP

Observamos que efectivamente la VLP crece a medida que el objeto se aleja. Finalmente se enunciará las ventajas y desventajas encontradas asociadas a este método.

2.3.2.1 Ventajas

- **Precisión.** Si las dos variables necesarias son ciertas se calcula la relación prácticamente sin estimaciones, lo que resulta en medidas muy precisas.
- **Facilidad de cálculo.** Como se puede observar únicamente se realiza una división entre una constante y una variable.

2.3.2.2 Desventajas

- **Requisitos.** Se debe conocer alguna medida de longitud real del objeto de interés.

- **Versatilidad.** La característica especificada para el cálculo no puede variar a través del tiempo, por ejemplo una circunferencia que se achata por efecto de la presión, ver figura 2.5. También es necesario que la característica mantenga su alineación con las cámaras, por ejemplo si se tratara de una línea en una hoja de papel, el objeto podría cambiar sus propiedades, por ejemplo, si el papel se inclinase, ver figura 2.6.

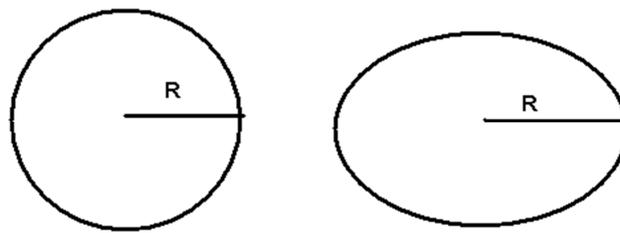


Figura. 2.5. Cambio de características por propiedades físicas

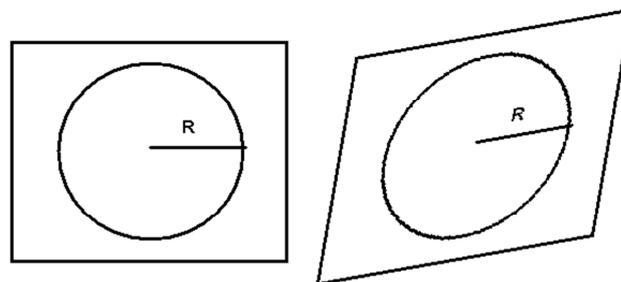


Figura. 2.6. Cambio de características por alineación

- **Relación con los objetivos.** A pesar de que la VLP guarda una estrecha relación con la distancia a la que se encuentre el objeto, en este método no se utiliza este hecho, incluso podría utilizarse si no se tratará de un sistema de visión estéreo.

2.3.3 Método de Aproximación por distancia

Al igual que el método anterior, su principal objetivo es encontrar la VLP. Basándose principalmente en el hecho de que esta variable evidentemente mantiene una relación con la distancia a la que se encuentren los objetos de una escena, se propone encontrar la función que relacione estas dos variables, VLP y

profundidad, a través de un método experimental basado en toma de muestras y en el método anterior. A continuación se describe el método.

- A distancias conocidas de separación de los dispositivos de adquisición, se toman muestras de el tamaño de una característica conocida en pixeles.
- Se obtienen los valores individuales de VLP, dividiendo la longitud de característica real entre el valor en pixeles.
- Se grafica la relación VLP en función de la distancia.
- Se encuentra estadísticamente la ecuación que más se ajuste a la curva.

De esta forma se obtiene una función particular que para un valor de distancia obtiene un el valor de VLP. Finalmente se enunciará las ventajas y desventajas encontradas asociadas a este método.

2.3.3.1 Ventajas

La principal ventaja de este método se presenta en la versatilidad, al estimar la ecuación, la variable VLP se puede hallar en función de la distancia, por ello no es necesario especificar un dato real de longitud. Además al no depender de este dato el inconveniente de que el objeto tenga que estar alineado a las cámaras se supera al igual que la necesidad de que el procesamiento de señales tenga que conocer la característica de longitud para realizar el cálculo de VLP, si bien es cierto identificar el radio o la arista de un objeto es sencillo para el procesamiento de imágenes existen imágenes donde no existen particularidades de este tipo, de esta forma se facilita la posibilidad de trabajar con varios tipos de objeto.

Por otro lado se aprovecha el conocimiento de la profundidad otorgado por el sistema estéreo, lo que tiene concordancia con los objetivos del proyecto.

2.3.3.2 Desventajas

La principal desventaja consiste en la necesidad de usar una función experimental, por lo tanto la precisión y confiabilidad de los resultados nunca podrá ser tan alta en comparación al método anterior, cabe recalcar en esta sección que la curva depende de la resolución a la que se desee trabajar ya que a diferentes resoluciones la cantidad de píxeles usados para representar las misma imágenes es diferente.

2.3.4 Selección del algoritmo de ubicación en el plano imagen

Se ha concluido utilizar el método de aproximación por distancia, ya que realmente es una extensión del método de característica conocida que brinda versatilidad ampliada y se relaciona estrechamente con el conocimiento de la profundidad, el cuál es una de las bases de esta investigación. Una vez definida VLP resulta sencillo encontrar los componentes del plano imagen, al multiplicar las coordenadas del objeto con la variable VLP.

2.4 ARQUITECTURA DEL SISTEMA DE VISIÓN

Una vez definidos los procesos que serán utilizados para la obtención de los atributos de interés, es necesario definir los elementos del sistema de visión. A continuación se describirán estos elementos y su lugar dentro de la arquitectura del sistema.

2.4.1 Diagrama en bloques de elementos

Los elementos que se van a utilizar para el sistema de visión son:

ELEMENTOS	
Dispositivo de Captura	iLook 300 marca Genius sensor CMOS 300kpixel
Conversor A/D	Inmerso en circuitería de la cámara
Memoria de Video	Inmersa en la circuitería de la cámara
Procesador	Pentium IV
Memoria de procesamiento	512 Mb en Ram
Monitor	Samsung CRT

Tabla. 2.1. Dispositivos de la arquitectura del sistema

Como se puede observar la arquitectura consta básicamente de un PC con procesador Pentium IV para el procesamiento, dos cámaras web para la captura y un monitor VGA para visualización. Los elementos de conversión A/D y memoria de video se encuentran inmersos en la circuitería de la cámara, la memoria de video en una cámara web es mínima y únicamente se utiliza para un tratamiento básico de las imágenes y su traslado mediante el puerto USB a la PC. Por otro lado el PC debe contar con al menos 128 Mb de memoria RAM para que el controlador de la cámara maneje las imágenes entregadas por ella. En la figura 2.7 se puede observar la disposición de estos elementos.

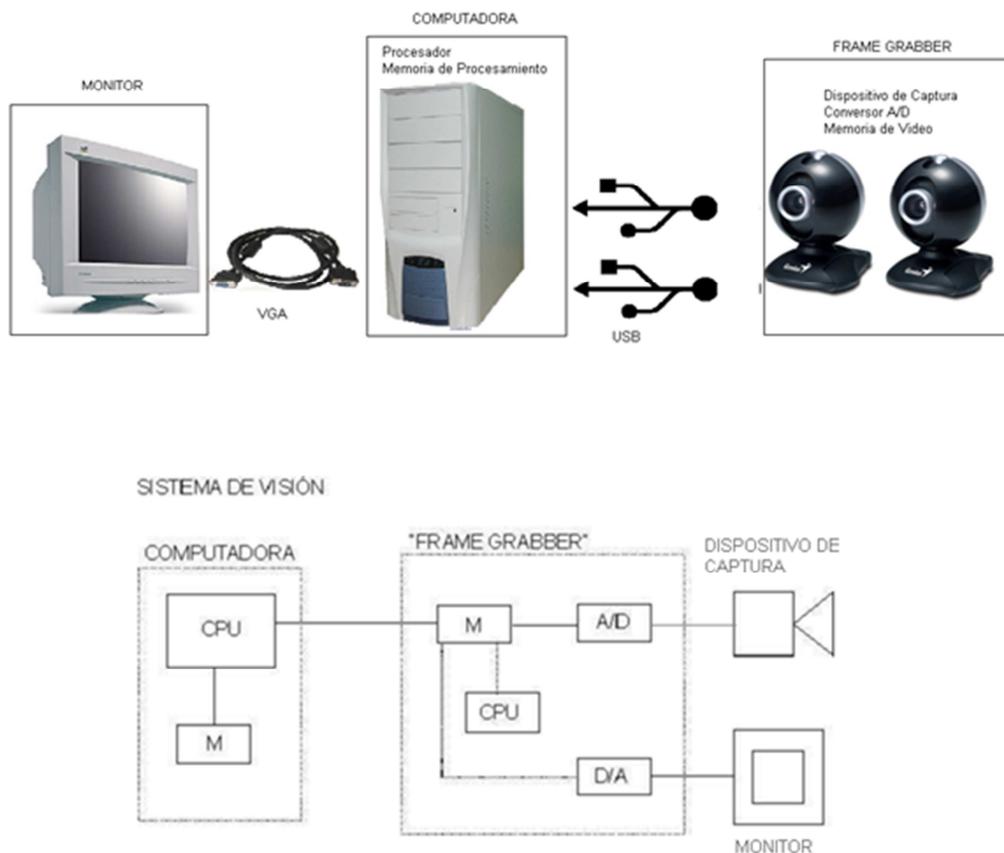


Figura. 2.7. Arquitectura del sistema de Visión

2.4.1.1 Dispositivos de Captura

Genius Ilook 300	
Tipo de sensor de imagen	Sensor de imagen CMOS 300k pixel
Tipo de lente	Enfoque manual
Interfaz	Compatible con USB 1.1/1.0
Formato del archivo	JPEG/WMV
Resolución	640x480,352x288,320x240,176x144,160x120 pixeles
Cuadros por segundo	CIF / 30 fps VGA / 30 fps
Angulo de Visión	Arriba: 25 grados / Abajo: 25 grados

Tabla. 2.2. Especificaciones del Dispositivo de Captura

2.4.1.2 Unidad central de Proceso

MOTHERBOARD	
Tipo de CPU	Intel Pentium 4 630, 3000 MHz (15 x 200)
Nombre del Motherboard	Intel Vanguard D915GLVG (2 PCI, 1 PCI-E x1, 4 DDR DIMM, Audio, Video, LAN)
Chipset del Motherboard	Intel Grantsdale-G i915GL
Memoria del sistema	502 MB (DDR SDRAM)
Memoria soportada	1024 MB
Slots de memoria	4 (2 libres)
Tipo de BIOS	AMI (05/30/05)
Puertos de Comunicación	Puerto de Comunicaciones (COM1) Puerto de Impresora (LPT1)
CACHÉ	
Caché de pista L1	Instrucciones de 12KB
Caché de datos L1	16KB
Caché L2	2 MB

Tabla. 2.3. Características CPU

2.4.1.3 Video

MONITOR	
Marca – Modelo	Samsung – SyncMaster 551v
Resolución máxima	1280x1024 (32 bit de profundidad)
Conector	Intel Grantsdale-G i915GL
TARJETA DE VIDEO	
Placa de Video	Intel GMA 900
Aceleradora 3d	Intel GMA 900
Velocidad de núcleo	333 Mhz
Memoria	Hasta 128 de memoria compartida

Tabla. 2.4. Características de video

2.5 ALGORITMO IMPLEMENTADO

El Sistema de Visión Estereoscópica para el presente proyecto fue implementado en la plataforma de programación grafica llamada LabView de la empresa National Instruments creado en 1976.

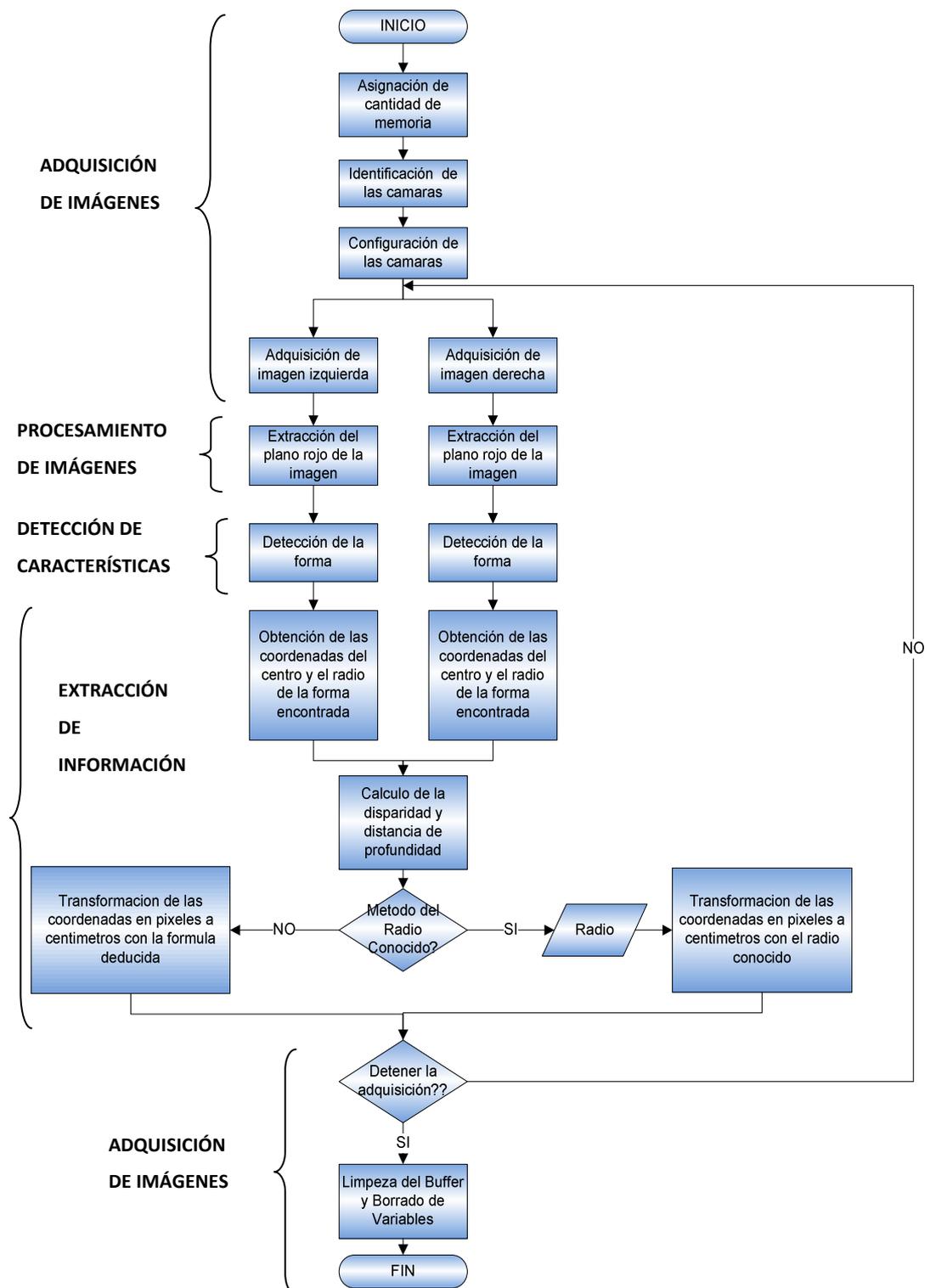
Inicialmente fue concebida para realizar tareas de control a instrumentos de medición pero que actualmente es utilizada para fines tan diversos como automatización industrial, diseño, adquisición de datos, visión artificial, entre otros. Una de las ventajas del LabVIEW, es que puede ser usado tanto por programadores expertos, como por usuarios intermedios, ya que es un lenguaje gráfico donde las instrucciones están “dibujadas” y no escritos en líneas de código, así, se pueden comprender fácilmente.

Los programas realizados en LabVIEW se llaman instrumentos virtuales “VI”, ya que tienen la apariencia de instrumentos reales, sin embargo, poseen analogías con funciones provenientes de lenguaje de programación convencionales.

Los VI están conformados por dos interfaces: el panel frontal y el diagrama de bloques. El panel frontal es la interfaz con el usuario, ya que se pueden ingresar datos y visualizar resultados. En el diagrama de bloques se encuentra todo el código fuente del VI. Este panel frontal (interfaz con el usuario) se crea al mismo tiempo que se va desarrollando el programa de forma automática lo que facilita de forma significativa la creación de programas en menos tiempo.

2.5.1 Diagrama de Flujo

El programa implementado básicamente se basa en cuatro pasos como la adquisición de imágenes, el procesamiento de las imágenes, la detección de características y por último la extracción de información.

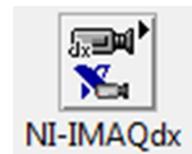


2.5.2 Adquisición de imágenes

El Software Vision Acquisition es un software controlador para adquirir, visualizar, registrar y monitorear imágenes desde varios tipos de cámaras. Con un juego de funciones fáciles de usar y programas ejemplo, con la posibilidad de crear rápidamente aplicaciones en NI LabVIEW, C, C++, Visual Basic y Visual Basic .NET.

Labview soporta una gran cantidad de cámaras ya sean estas webcams, microscópicas, escáneres y muchas otras, todas estas divididas en 2 grandes grupos como lo son las cámaras análogas y digitales.

Este software incluye NI-IMAQ, un controlador para adquirir desde cámaras analógicas, digitales paralelas y Camera Link así como NI Smart Cameras. El software también tiene el controlador NI-IMAQdx para adquirir desde dispositivos GigE Vision, cámaras IEEE 1394 compatibles con IIDC y dispositivos USB compatibles con DirectShow como cámaras, webcams, microscopios, escáner y varios productos de imagen.



En la figura 2.8 se muestra como se uso en el presente proyecto las herramientas de adquisición de imágenes de Labview.

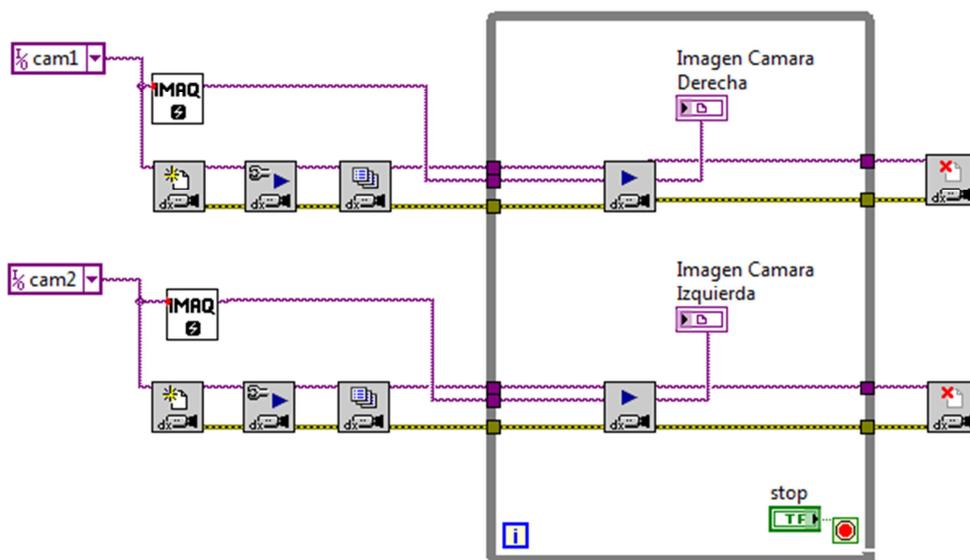


Figura. 2.8. Programa para la adquisición de imágenes

Los bloques que permiten la adquisición de datos se describirán brevemente a continuación:

Imaq Create. Contiene un puntero de información para buffer, que inicialmente se encuentra vacío. La adquisición de la imagen asigna la cantidad correcta de memoria para la adquisición. Si usted va a procesar la imagen, puede ser que necesite para proporcionar un valor para el Tamaño del Borde (*Border Size*). Tamaño del borde (*Border Size*) es la anchura, en píxeles, de una frontera creada en torno a una imagen. Algunas Funciones de procesamiento de imágenes, tales como el etiquetado o en la morfología, requieren de un borde.

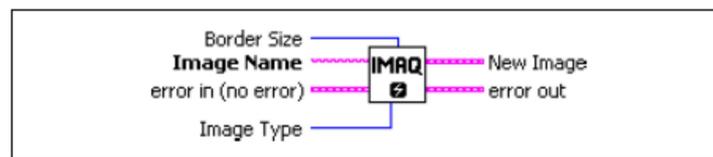


Figura. 2.9. Bloque Imaq Create

Imaqdx Open Camera. Para adquirir imágenes con funciones de alto o de bajo nivel, por primera vez se debe abrir una sesión de cámara. Una sesión de cámara es un proceso seguro para manejar a una cámara.

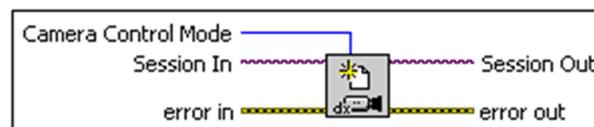


Figura. 2.10. Bloque Imaqdx Open Camera

El controlador utiliza esto para identificar la cámara la cual queda lista para aplicar funciones de NI-IMAQdx. Se puede abrir tantas sesiones de la cámara como cámaras conectadas a su sistema.

Al abrir una sesión, es necesario especificar dos parámetros: nombre de la cámara y el modo de control de la cámara. Cuando una aplicación termina con la cámara, se llame a la función *Close Camera* para cerrar la sesión de cámara.

Nombre de la cámara. NI-IMAQdx referencia a todas las sesiones de la cámara mediante un nombre. El controlador crea nombres por defecto para cada cámara en el sistema en el orden en que las cámaras están conectadas.

La manera de nombramiento de las cámaras se muestra en la tabla 2.5.

Nombre de la Cámara	Cámara Instalada
cam0	Dispositivo 0
cam1	Dispositivo 1
...	...
camn	Dispositivo n

Tabla. 2.5. Nombramiento de las Cámaras

Modo de control de la cámara. El parámetro control de la cámara tiene dos opciones controlador y oyente. La opción por defecto es controlador el cual maneja la cámara y recibe datos de video. El oyente sólo recibe datos de vídeo.

Imaqdx Configure Grab. Después de abrir la cámara, se debe configurar la cámara para la adquisición especificando los siguientes parámetros: si la adquisición es de una sola vez o continua, y el número de búferes internos para su uso.

Imaqdx Configure Grab. Configura y comienza la adquisición. Realiza una adquisición que se repite continuamente en un anillo de búferes.

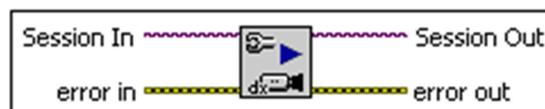


Figura. 2.11. Bloque Imaqdx Configure Grab

Imaqdx Grab. Inicia una adquisición de imágenes continua de alta velocidad a uno o más búferes internos.

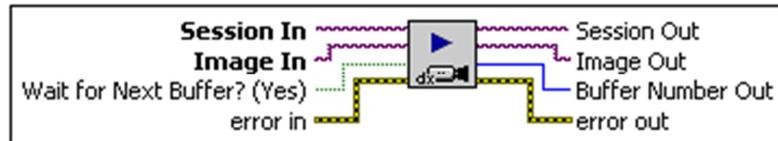


Figura. 2.12. Bloque Imaqdx Grab

La figura 2.13 ilustra el orden de programación típico de una adquisición de alto nivel.

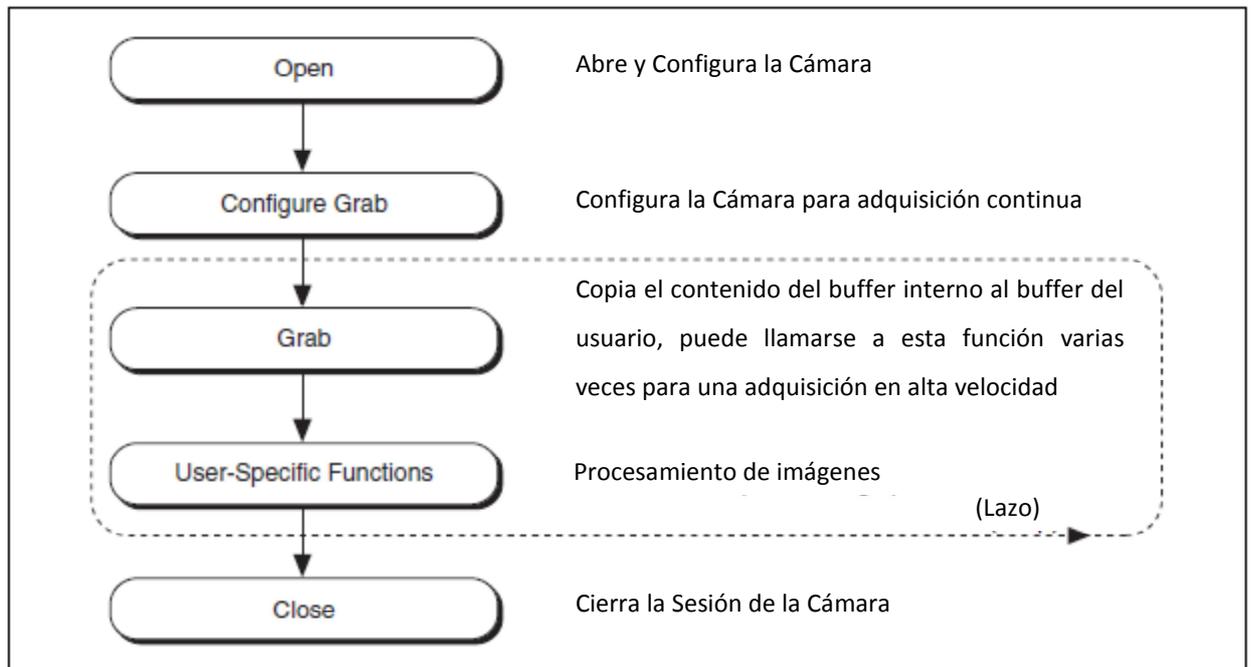


Figura. 2.13. Flujo para la adquisición de alto nivel

2.5.3 Procesamiento de la imagen

En esta etapa del algoritmo se pretende por medio de un conjunto de técnicas facilitar la búsqueda de información dentro de la imagen digital obtenida anteriormente. Primeramente se manipula los planos de color de la imagen a una escala monocromática esto quiere decir que la información de la imagen pasa de estar guardada en las tres matrices ($3 \times M \times N$) que comúnmente se utiliza en el formato RGB a solamente una matriz ($1 \times M \times N$). Las ventajas de utilizar esta técnica no solo limitan el ahorro de memoria, compresión de imágenes y mejora de tiempos de transmisión o muestra de imágenes.

El bloque de comando que se utilizó para este objetivo es el siguiente:

IMAQ ExtractSingleColorPlane. Para cumplir este objetivo nos valemos de la herramienta color de LabView la cual permite manipular colores y planos de color de una imagen. En la tabla 2.6 se describen los planos que se puede extraer de la imagen:

Red (0)	(Default) Extracts the red color plane
Green (1)	Extracts the green color plane
Blue (2)	Extracts the blue color plane
Hue (3)	Extracts the hue color plane
Saturation (4)	Extracts the saturation color plane
Luminance (5)	Extracts the luminance color plane
Value (6)	Extracts the value color plane
Intensity (7)	Extracts the intensity color plane

Tabla. 2.6. Planos de color

La figura 2.14 muestra el código que se utilizó para lograr la extracción del plano de color rojo en las imágenes obtenidas anteriormente, izquierda como derecha.

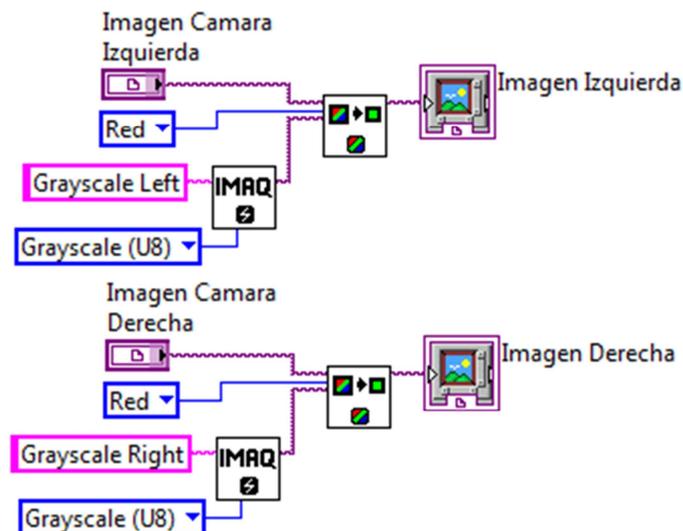


Figura. 2.14. Extracción del plano de color de las imágenes

2.5.4 Detección de características

Una vez que la imagen se encuentra en un solo plano de color, podemos procesarla para rastrear o buscar ciertas características de la misma.

Labview posee algunas opciones de búsqueda de características en una imagen como son: La detección de patrones, sub imágenes, plantillas, colores y formas definidas (círculos, elipses, líneas y rectángulos).

Si bien es cierto la detección por patrones es la mejor de las opciones ya que esta encuentra objetos específicos evitando confusiones con otros parecidos, lamentablemente esta técnica ocupa mucho tiempo de procesamiento que para el objetivo de este proyecto es un problema, ya que se quiere alcanzar o aproximar el tiempo de respuesta del sistema a un tiempo real, esto quiere decir de ejecución casi instantánea.

La otra opción que es la de detección de color, la cual posee un tiempo de procesamiento conveniente para el proyecto pero no es tan fiable en su búsqueda por motivo de que el objeto se va a mover o desplazar a lo largo de un ambiente con situaciones imprevistas de iluminación variando así los matices de color del patrón de búsqueda, es descartada.

La detección de formas simples es la opción más acertada para este proyecto ya que esta no tiene tiempos altos de procesamiento y no es dependiente de la iluminación.

Una vez discernidas las opciones de búsqueda procedemos a encontrar las formas circulares y sus coordenadas dentro de la imagen. El bloque que permite realizar este objetivo es el siguiente:

IMAQ Detect Circles. Este comando permite detectar círculos dentro de una imagen, ver figura 2.15. Se debe configurar una serie de parámetros, entre los cuales están:

- **Modo de extracción.** Especifica cómo el VI identifica las curvas en la imagen.
- **Borde del umbral.** Especifica el contraste mínimo para que un punto que debe tener para que comience una curva. Los valores válidos oscilan entre 1 y 255. El valor por defecto es de 75.
- **Tamaño del borde del filtro.** Especifica el ancho del borde del filtro que el VI utiliza para identificar las curvas en la imagen
- **Longitud mínima.** Especifica la longitud en píxeles, de la más pequeña curva que desea que el VI de identificar. El VI ignora cualquier curva que tiene una longitud inferior a este valor. El valor por defecto es 25.
- **Tamaño paso de búsqueda en filas.** Especifica la distancia en la dirección “y”, entre las filas de la imagen que el algoritmo inspecciona los puntos de la curva. Los valores válidos oscilan entre 1 y 255. El valor por defecto es 15.
- **Tamaño paso de búsqueda en columnas.** Especifica la distancia en la dirección “x”, entre las columnas de la imagen que el algoritmo inspecciona los puntos de la curva. Los valores válidos oscilan entre 1 y 255. El valor por defecto es 15.
- **Número máximo de brecha final.** Especifica la distancia máxima en píxeles, entre los extremos de una curva que el VI identifica como una curva cerrada. Si el espacio es mayor que este valor, el VI identifica la curva como una curva abierta. El valor por defecto es 10.
- **Cerrado.** Especifica si se identifican solo las curvas cerradas en la imagen. Si el valor predeterminado es FALSO, especifica que el VI debe identificar tanto las curvas abiertas y cerradas.
- **Subpíxeles.** Si este es configurado como verdadero los subpíxeles son ignorado por el VI.
- **Rotación.** Especifica si se busca o no versiones de la forma rotada o girada
- **Escala:** especifica si se busca o no versiones a escala de la forma.
- **Oclusión:** especifica si se busca o no versiones de la forma obstruida.
- **Radio mínimo:** especifica el radio mínimo de los círculos a detectar.
- **Radio máximo:** especifica el radio máximo de los círculos a detectar.

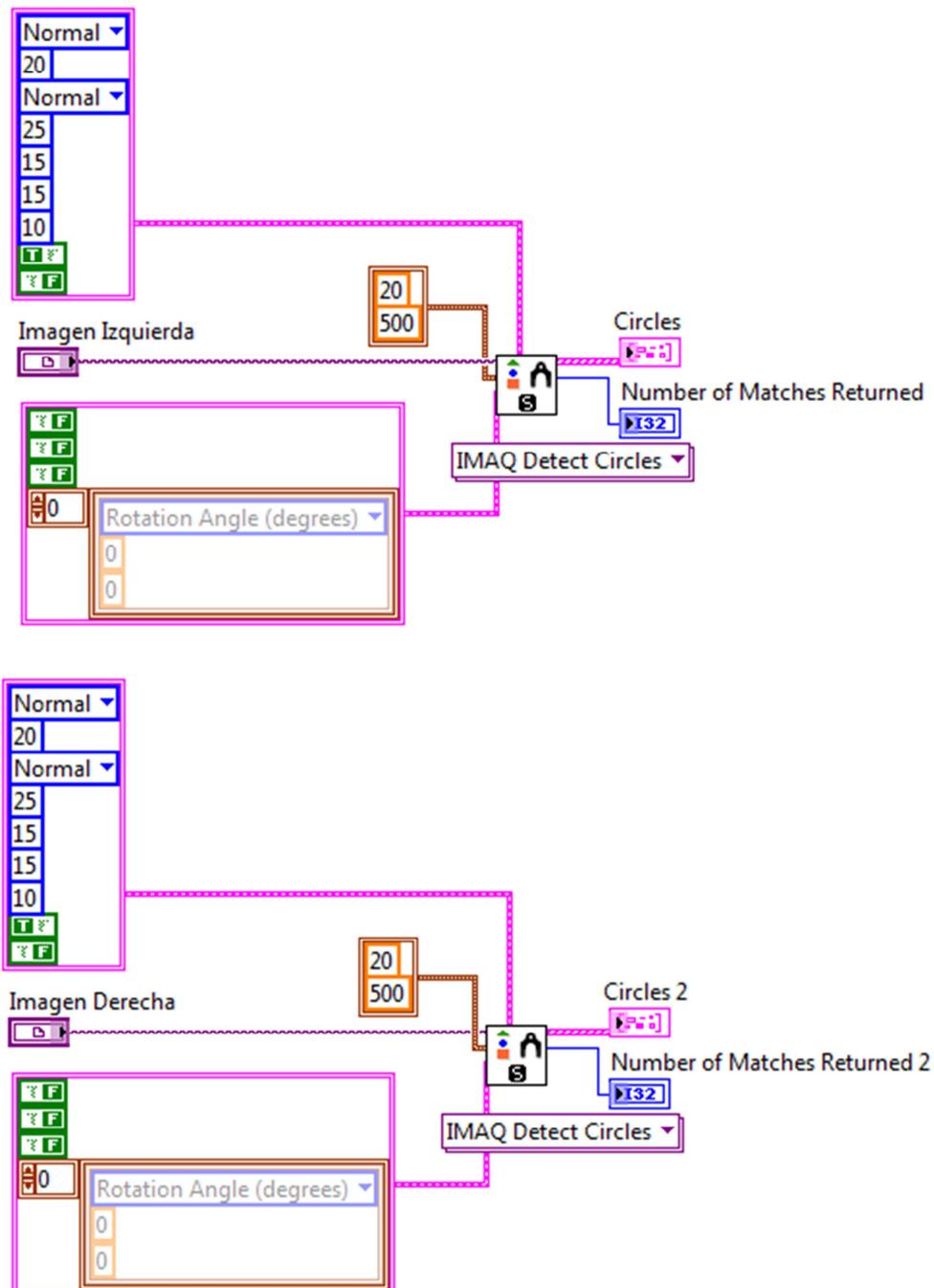


Figura. 2.15. Detección de círculos

2.5.5 Extracción de información

Una vez localizado el objeto, Labview nos entrega una serie de datos esenciales para lograr las coordenadas espaciales reales de la forma que se pretende seguir. Entre la información necesaria que obtenemos por medio de esta herramienta son las coordenadas y el radio en pixeles de la forma encontrada en

la cámara derecha, como en la izquierda. En la figura 2.16 se muestra el programa que permitió la extracción de información.

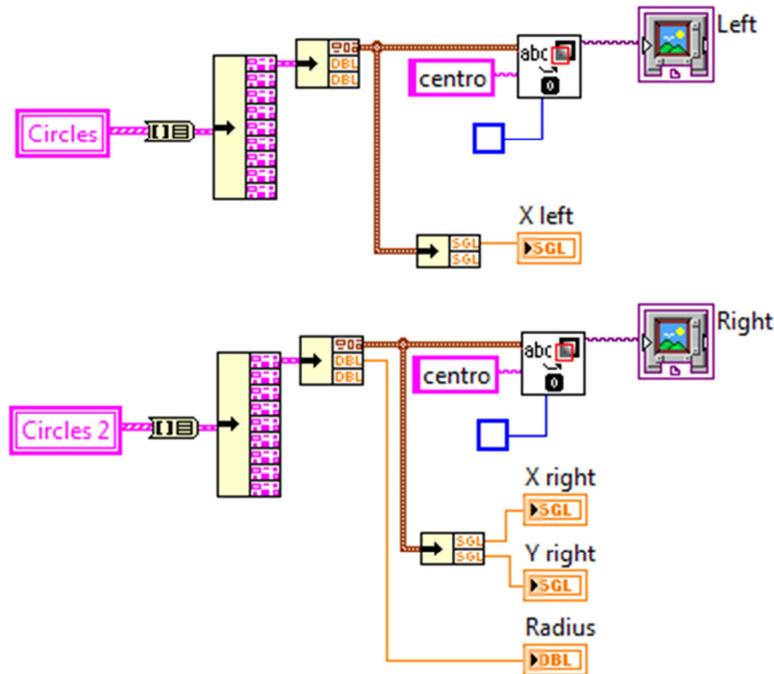


Figura. 2.16. Extracción de información relevante

Los datos obtenidos nos van a servir para el cálculo de la disparidad. La distancia de separación de las cámaras, la distancia focal, la constante de proporcionalidad y la disparidad antes calculada permiten el hallazgo de la distancia real de profundidad del objeto a las cámaras. La figura 2.17 muestra el código del programa para el hallazgo de estos datos.

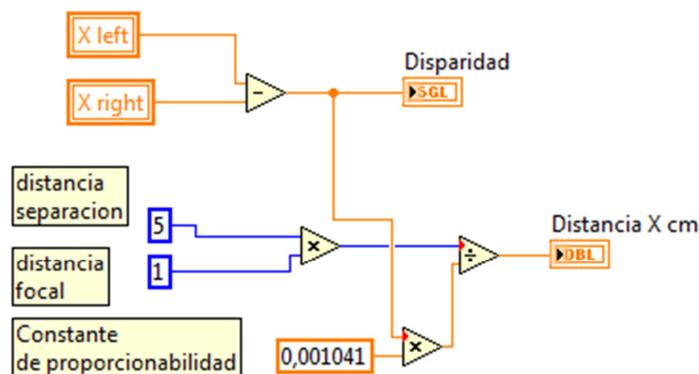


Figura. 2.17. Cálculo de la disparidad y la distancia

Para el cálculo de las dos coordenadas sobrantes nos valemos de la posición (en pixeles) del objeto en la imagen de la cámara derecha, de manera que se fija un *setpoint*, el cual va a ser la posición del objeto centrado a las dos cámaras en la mínima distancia en que lo visualizan. La figura 2.18 muestra el *setpoint* y los desplazamientos que el objeto debe realizar para que este alcance la alineación con el punto de referencia.

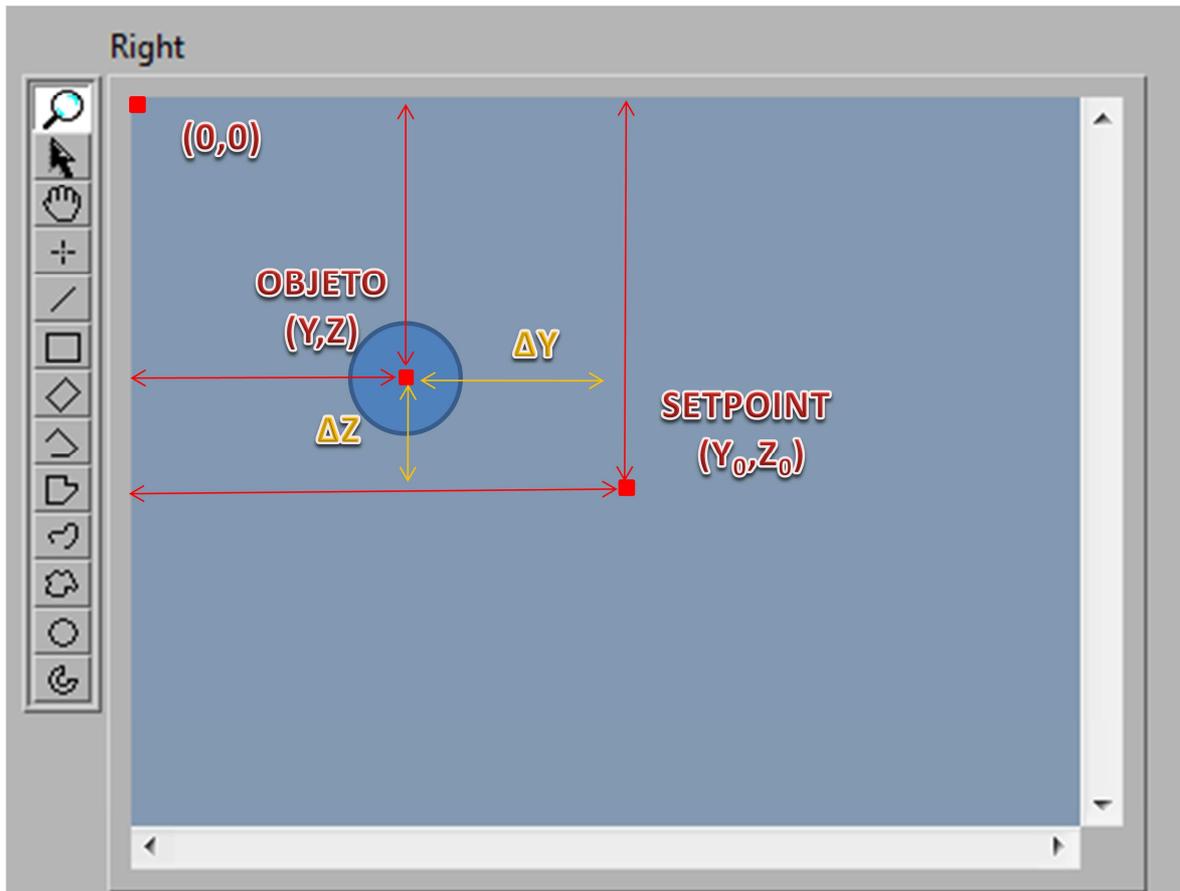


Figura. 2.18. Descripción de la ubicación de setpoint

El sistema de visión va a permitir el cálculo de la distancia horizontal como vertical del objeto por medio de dos técnicas. Las técnicas implementadas tienen como objetivo la traducción de los desplazamientos del objeto en relación al setpoint ya calculados en pixeles a unidades reales de medida como en este caso centímetros.

La primera técnica es la de la característica conocida la cual necesita el ingreso al programa de la medida del radio del objeto a detectar

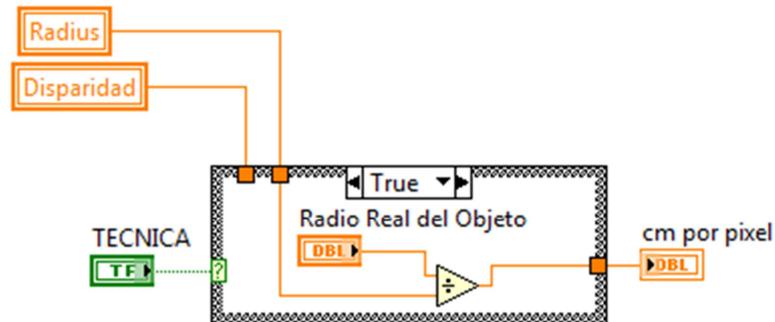


Figura. 2.19. Técnica del Radio conocido

La segunda técnica logra el objetivo por medio de la técnica de aproximación por distancia la cual usa una función estimada en base a la distancia.

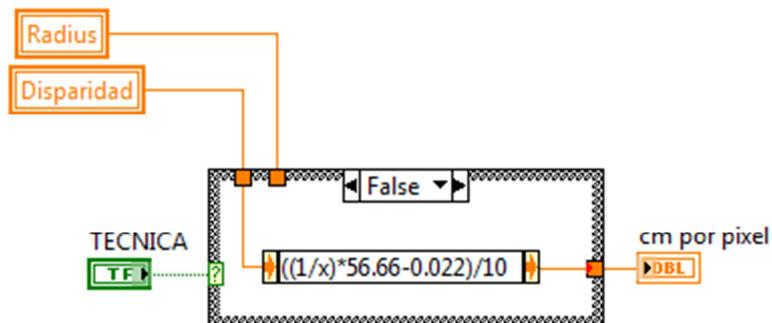


Figura. 2.20. Técnica de la Función experimental

Una vez encontrado la variable VLP que permite la traducción pixeles a centímetros por cualquiera de las dos técnicas se procede a transformar los desplazamientos a centímetros, como muestra la figura 2.21.

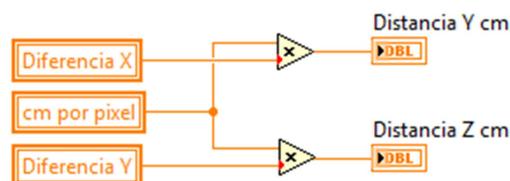


Figura. 2.21. Transformación de las distancias en pixeles a centímetros

CAPÍTULO 3

POSICIONAMIENTO DEL BRAZO ROBÓTICO

3.1 INTRODUCCIÓN

En este capítulo se explicará el procedimiento de diseño y la implementación del algoritmo de posicionamiento, mediante este el brazo robótico realizará la tarea de seguimiento de un objeto. Recordando el capítulo anterior, tenemos como punto de partida la información de ubicación espacial del objeto a seguir respecto a las cámaras.

En primer lugar, se analizará la manera en la que los datos de ubicación espacial del objeto serán transmitidos hacia el controlador robótico C500, en segundo lugar se definirá el problema del movimiento del manipulador, se describirá la estrategia de movimiento y los comandos a utilizarse. Luego se describirá el algoritmo de posicionamiento implementado, sus características y limitaciones.

3.2 COMUNICACIÓN E INICIALIZACIÓN

3.2.1 Comunicación

El controlador C500 como se describió en el primer capítulo posee varias maneras de comunicación e interface con periféricos y dispositivos externos. En nuestro caso el controlador debe comunicarse con el Sistema de Visión implementado en un computador, para este objetivo la mejor opción de intercambio de información es por el puerto de comunicaciones del controlador el cual transmite los datos de manera serial RS232.

El puerto de comunicaciones puede actuar de dos diferentes modos, el modo interactivo y el modo de programa. El modo de programa utiliza el protocolo ACI para descargar y subir datos a la memoria del controlador, este modo permite la ejecución de una secuencia de comandos de forma automática. Por otro lugar el modo interactivo, permite la ejecución de comandos en forma manual, esto quiere decir que el robot ejecuta el comando después de ingresado. Para nuestro propósito utilizaremos el modo interactivo ya que este nos va a permitir mandar comandos desde el programa en LabView para su ejecución inmediata y a la vez el control de flujo del algoritmo.

3.2.2 Inicialización

El controlador para comenzar la comunicación establece una secuencia de inicialización. Para obtener estos códigos se realizo ingeniería inversa ya que la información sobre esto es limitada.

Con la ayuda de un programa monitor del puerto serial se evaluó el momento en el que el programa *RobComm* inicializa al controlador, de esta manera se obtuvo el siguiente resultado que se muestra en la figura 3.1

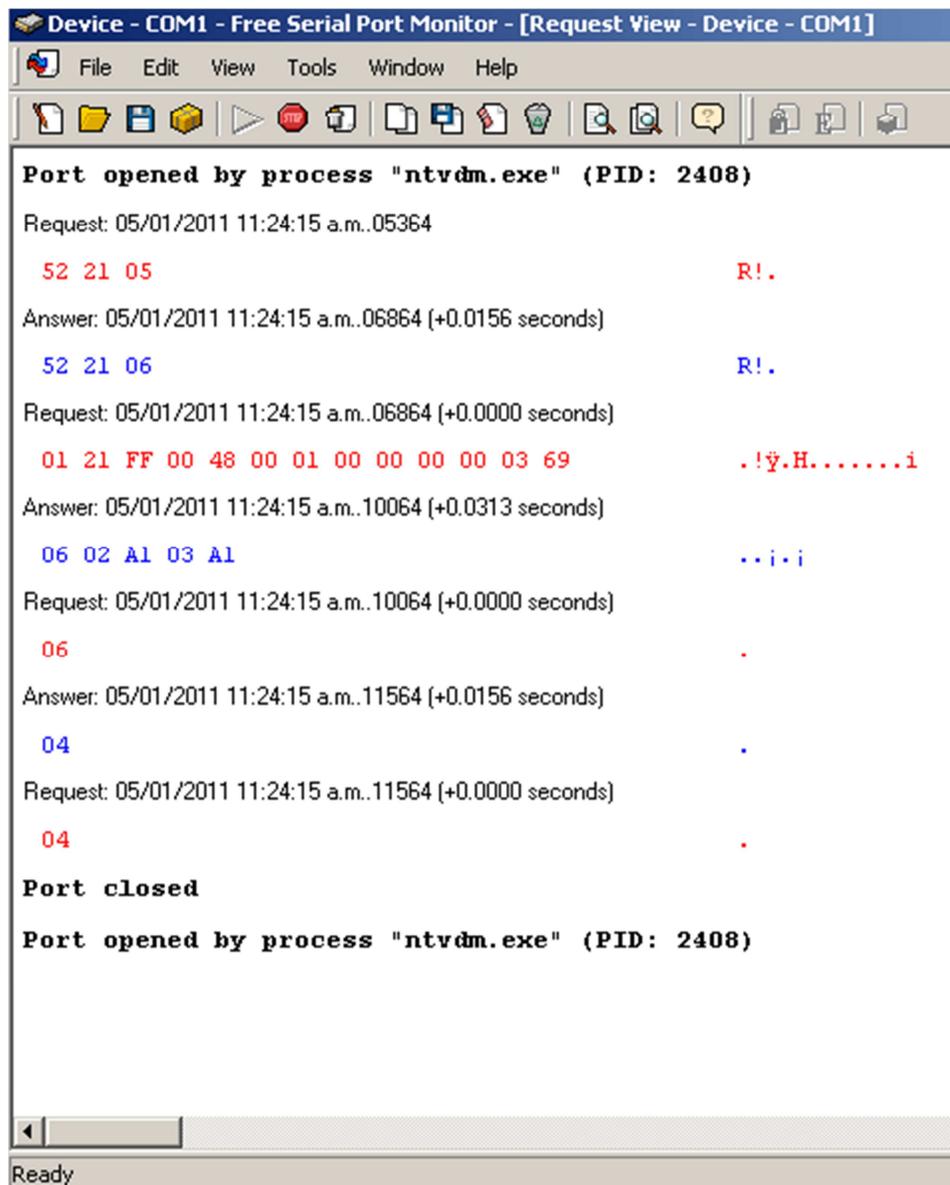


Figura. 3.1. Ventana del Programa Monitor del Puerto Serial

Como se observo, son códigos hexadecimales que se envían en bloques separados por diferentes tiempos.

Una vez obtenidos estos códigos y con la información del manual del controlador podemos concluir la siguiente secuencia de inicialización del controlador.

Secuencia de petición. El dispositivo máster entrega el siguiente código de tres bytes:

'R', slave ID + 20h, ENQ

El máster debe seguir esta secuencia con dos periodos de no transmisión. Esto garantiza que el esclavo haya leído la secuencia de pedido, y evita coincidencia en la comunicación con otro dispositivo esclavo. Cuando el esclavo ha determinado que es una petición válida, este responderá lo siguiente:

'R', slave ID + 20, ACK

Una vez leída esta respuesta por el máster, un enlace de comunicación se ha establecido.

El número de identificación del dispositivo esclavo es cualquiera entre 0 y 7F hexadecimal (0 y 127 decimal). El programador puede configurar este número desde RAPL-II.

Al sumar un valor de 20h al número del esclavo proporciona seguridad para que el segundo byte de la cadena no se parezca a un código de control de cualquier tipo. De esta manera el dispositivo esclavo puede distinguir la secuencia de pedido y establecer la comunicación rápidamente.

El maestro tratará de contactar con el esclavo limitada número de veces antes de abandonar la comunicación. Si por alguna razón el esclavo no responde a la secuencia, puede que el esclavo requerido no esté presente en el enlace, o el rango de los baudios este configurado incorrectamente o existe un problema con la escritura del interface.

Carácter	Código Hexadecimal	Código Decimal
SOH	01	01
STX	02	02
ETX	03	03
EOT	04	04
ENQ	05	05
ACK	06	06
NAK	15	21
ETB	17	23
control-Z	1ª	26

Tabla. 3.1. Códigos de control ASCII

Secuencia de cabecera. El bloque de cabecera consiste en una descripción de los datos que van a tener lugar en la transferencia. La cabecera se describe a continuación en la tabla 3.2:

	Código Hexadecimal	Descripción
1	SOH	Carácter de inicio de la cabecera
2	ID ESCLAVO + 20H	Identificación del Esclavo
3	ID MASTER + 20H	Identificación del Máster
4	LECTURA/ESCRITURA	Indicador de lectura o escritura de datos
5	TIPO DE MEMORIA	Tipo de acceso de memoria
6	NUMERO DE BLOQUES COMPLETOS	Número de bloques completos (128-bytes) a ser transferidos
7	NUM. BYTES EN EL ULTIMO BLOQUE	Numero de bytes en el último bloque
8	OFFSET DE MEMORIA DEL BYTE BAJO	Dirección de inicio de la memoria
9	OFFSET DE MEMORIA DEL BYTE ALTO	"
10	SEGMENTO DE MEMORIA DEL BYTE BAJO	"
11	SEGMENTO DE MEMORIA DEL BYTE ALTO	"
12	ETX	Fin de los caracteres de texto
13	LRC	Verificación longitudinal de redundancia

Tabla. 3.2. Descripción del contenido de la Cabecera

Todos los valores son expresados en notación hexadecimal a menos que se indique lo contrario.

El número de identificación del dispositivo esclavo es el mismo que aparece en la secuencia de petición. La identificación del máster debe ser '01' ya que es un sistema esclavo-máster.

El byte de lectura/escritura identifica la dirección de la transferencia de memoria. Una operación de escritura transferirá memoria desde el esclavo hacia el máster. El comando lectura es lo contrario. Este transfiere memoria del máster al esclavo.

El código '00h' es memoria de propósito general. Este código permite al programador leer o escribir en cualquier byte de la memoria 8086. Esto hace al ACI una potente herramienta (y potencialmente peligrosa) para trabajar.

Los próximos dos bytes de la cabecera identifican la cantidad de memoria a ser transferida. Esta información es codificada como el número de bloques de datos llenos a ser transferidos (128 bytes por bloque) y el número de bytes que se mantuvieron en el último bloque de datos (parcialmente llenos). El último bloque de datos no puede tener cero bytes. Si el número de bytes de datos puede ser dividido exactamente para 128, el último bloque contiene 128 (80h) bytes.

Para determinar el número de bloques y número de bloques anteriores se sigue lo siguiente: N es el número de bytes totales a ser transferidos, B es el número de bloques llenos a ser transferidos, n es el número de bytes restantes en el último bloque.

$$B = \frac{N}{128}$$

$$n = N \bmod 128$$

If n = 0 then do

$$n = 128$$

$$B = B - 1$$

end_if

La dirección de memoria identifica la dirección de inicio de la operación de transferencia de datos. Esto define, un formato de segmento/*offset* Intel, donde en la memoria del controlador del robot el dato será leído o escrito. El segmento y el *offset* serán definidos en formato *word*.

Detección de error en la transferencia del encabezado. El byte LRC es un carácter de verificación de redundancia longitudinal. Este es la suma de bytes desde el segundo hasta el onceavo del bloque de cabecera. El dispositivo esclavo suma los valores de los caracteres recibidos y acepta la cabecera como correcta solo si lee un byte LRC igual al calculado. Si el LRC no concuerda, la cabecera no es aceptada y el esclavo emite un carácter NAK para que el máster retransmita el bloque de cabecera. El máster solo retransmitirá tres veces antes de abortar el ciclo de comunicación.

Si el bloque de cabecera es aceptado, el dispositivo esclavo emitirá un carácter ACK como respuesta. El dispositivo esclavo espera los códigos correctos de control, SOH y ETX durante la transmisión. Cualquier desviación de este patrón causara un error y la comunicación será abortada. El tiempo de espera deberá estar entre el fin de la secuencia de petición y la aceptación del carácter SOH, y la longitud del tiempo requerido entre el envío de los caracteres SOH y ETX.

3.3 DESCRIPCIÓN DEL PROBLEMA DE MOVIMIENTO

3.3.1 Escenario previo

A continuación se listarán los hechos que conforman el escenario previo para determinar la estrategia de movimiento.

- El sistema de visión estéreo está en capacidad de entregar la información de ubicación espacial respecto al eje de la herramienta del objeto a seguir.
- El controlador C500 es capaz de interpretar comandos que incluyen la información entregada por el sistema de visión

- Se han analizado los comandos de movimiento y modificación del sistema de coordenadas que interpreta el controlador
- Se ha definido la ubicación física del elemento de adquisición de imágenes, y condicionado su orientación espacial, dicho elemento no cambiará rotará alrededor del eje X (ROLL - YZ) o del eje Y (PITCH - XZ).

3.3.2 Problema de Orientación

En la figura 3.2 se condensa el escenario del que se dispone, y a partir de ahí el problema de orientación, es necesario recalcar que únicamente, se representa el plano XY del mundo, ya que la herramienta (cámaras) únicamente sufrirá un cambio de orientación respecto dicho eje.

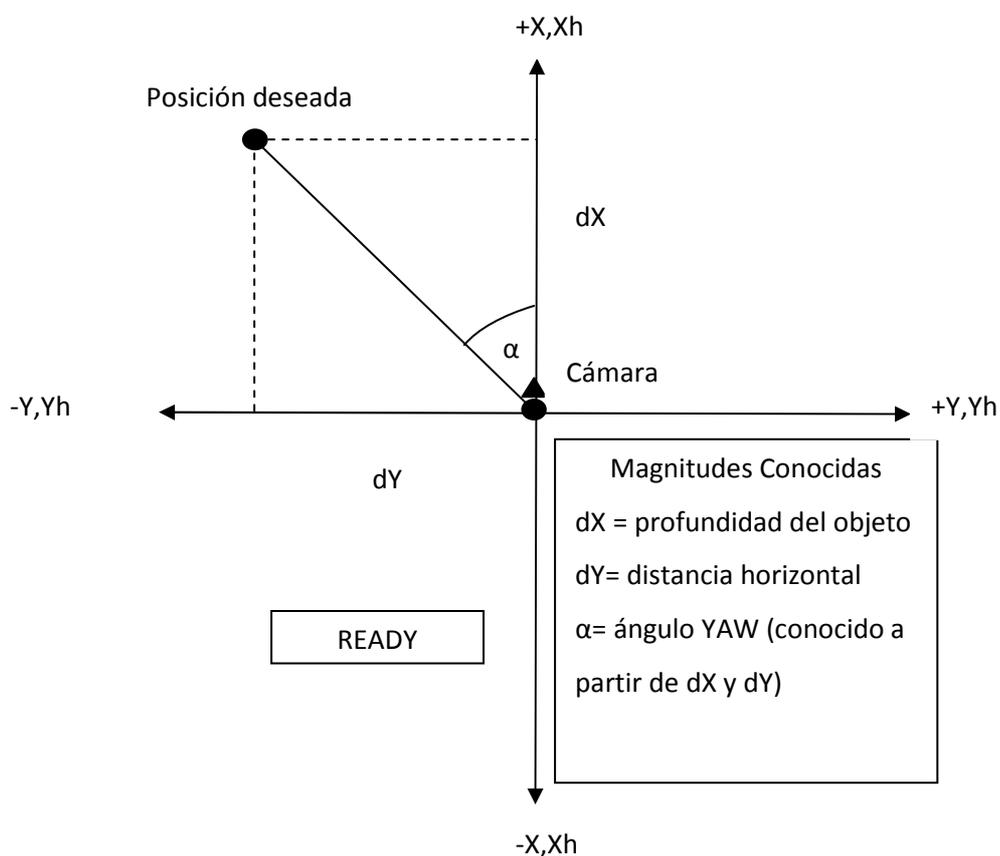


Figura. 3.2. Escenario inicial

En esta primera posición, llamada READY para el manipulador, se observa que el eje de la herramienta y el eje del mundo coinciden, utilizar cualquier

comando de movimiento que logre ubicar a la cámara en la posición deseada provocará un cambio en Y e implícitamente hará que el eje de la herramienta rote alrededor de Z. Como podemos ver en la figura 3.3.

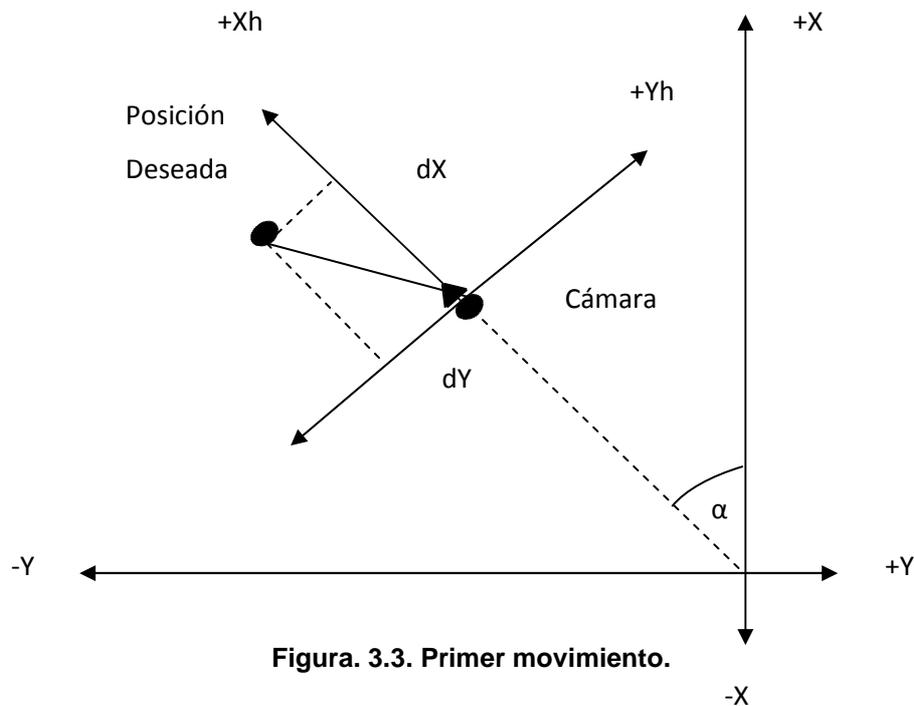


Figura. 3.3. Primer movimiento.

Ahora si se tiene una nueva posición deseada, observamos que el sistema de visión entrega su información en los ejes de la herramienta, si en este momento realizamos un comando de movimiento referido al sistema del mundo con dicha información, el manipulador no conseguirá posicionarse.

Con este análisis podemos definir los siguientes puntos:

- Si se utilizan comandos de movimiento del sistema de mundo, es necesario que la orientación del plano XY sea modificada con cada movimiento, de tal manera que el sistema de la herramienta este siempre alineado con el sistema del mundo.
- Si no se modifica la orientación del sistema, es necesario transformar las coordenadas de la herramienta al sistema del mundo, esto implica la utilización de matrices de transformación homogénea u otros métodos, o

utilizar comandos de movimiento que hagan referencia al sistema de coordenadas de la herramienta.

- Como se observa, la percepción de profundidad es siempre medida en el eje X de la herramienta, por lo tanto sería deseable que para cualquier movimiento de aproximación, primero se haya alcanzado una orientación hacia el objeto por parte de la cámara para evitar problemas de desenfocamiento.

3.3.3 Descripción de la estrategia propuesta

En base al escenario anterior y al análisis del problema fundamental de orientación se describirá, en qué forma se debería llevar a cabo la tarea de seguimiento.

A partir de la posición READY, figura 3.2, se conoce el ángulo YAW que tiene el objeto en relación al sistema coordenado, como primer paso, se realiza un movimiento que alinee al sistema de la herramienta o cámaras con el objeto como se puede ver en la figura 3.4.

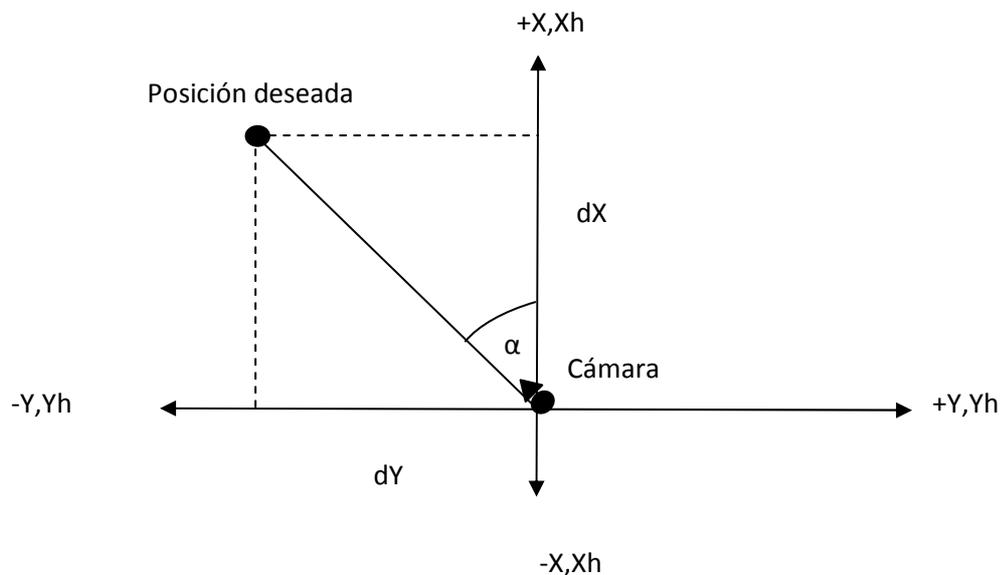


Figura. 3.4. Movimiento inicial

Luego se realiza la modificación del sistema coordenado con este ángulo, así:

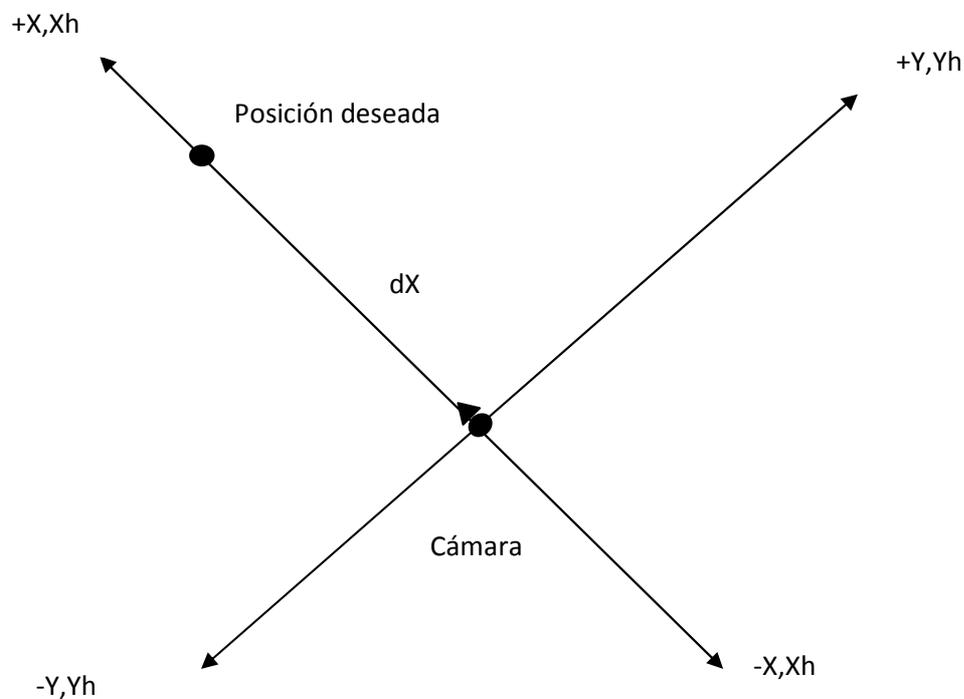


Figura. 3.5. Modificación del sistema con YAW

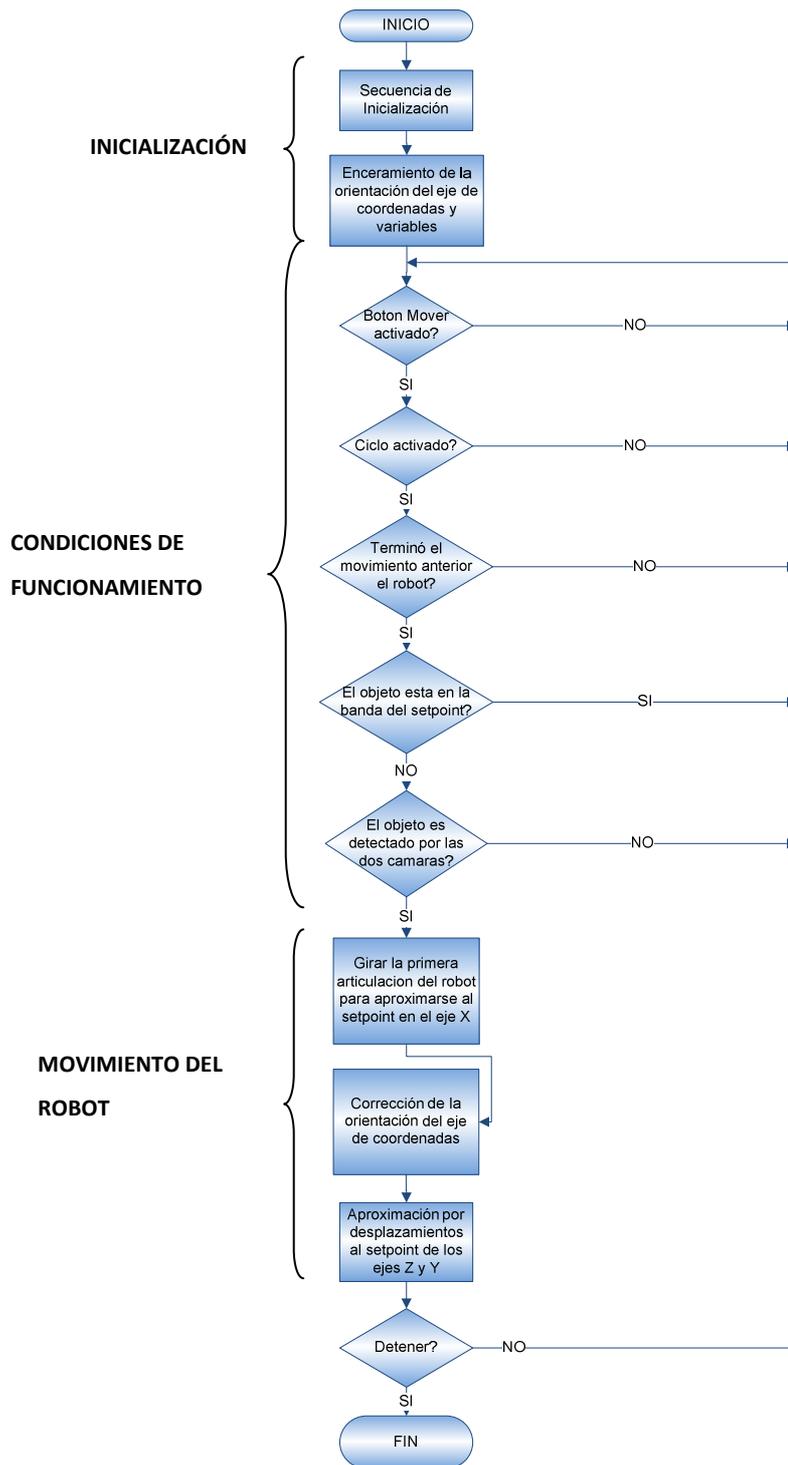
Entonces se procede a realizar los movimientos de profundidad y de distancia vertical, con lo que se consigue el posicionamiento y al hacerlo de manera repetitiva el posicionamiento.

3.3.4 Comandos a utilizarse

Para la estrategia de posicionamiento definida los comandos de utilización más directa son: JOINT, JOG y OFFSET, su funcionamiento, comandos complementarios y demás comandos de movimiento se pueden ver en el anexo B.

3.4 ALGORITMO IMPLEMENTADO

El programa implementado consta de tres partes: La inicialización, Las condiciones de funcionamiento y el movimiento del robot.



3.4.1 Inicialización

Se parte con la configuración del puerto serial que va a permitir la comunicación con el controlador C500. LabView posee la librería VISA (*Virtual Instrument Software Architecture*) la cual permite el control de VXI, GPIB, RS-232, y otros tipos de instrumentos.

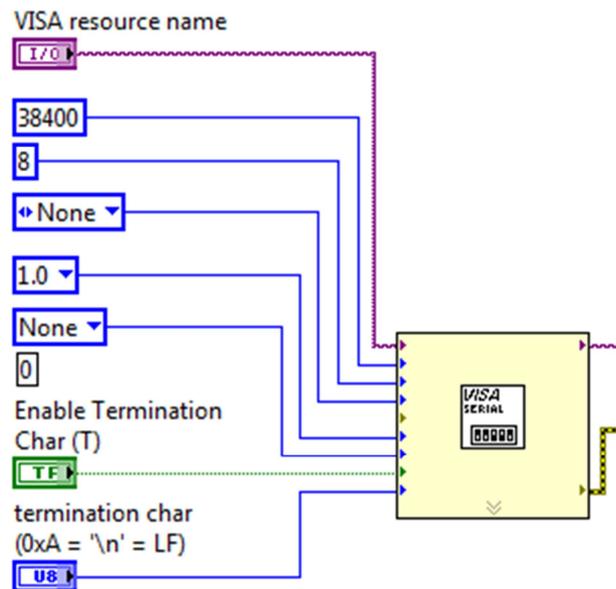


Figura. 3.6. Configuración e Inicialización del puerto Serial

Para inicializar y configurar el puerto serial se utilizó el comando de VISA *Configure Serial Port* mostrado en la figura anterior. Este comando permite configurar los siguientes parámetros:

- **VISA resource name.** Especifica el nombre del recurso abierto.
- **Baud rate.** Es la tasa de transmisión. El valor predeterminado es 9600.
- **Data bits.** Es el número de bits de los datos entrantes. El valor de los bits de datos se encuentra entre cinco y ocho. El valor predeterminado es 8.

- **Parity.** Especifica la paridad utilizada para cada trama que se transmite o se recibe. Esta entrada acepta los siguientes valores.

0	no parity (default)
1	odd parity
2	even parity
3	mark parity
4	space parity

- **Stop bits.** Especifica el número de bits de parada que se utilizan para indicar el final de una trama. Esta entrada acepta los siguientes valores.

10	1 stop bit
15	1.5 stop bits
20	2 stop bits

- **Flow control.** Establece el tipo de control utilizado por el mecanismo de transferencia.
- **Enable Termination Char.** Prepara el dispositivo serial para que reconozca una terminación por medio de un carácter.

Una vez abierta y configurada la transmisión serial se procede a mandar los códigos de inicialización del controlador C500 antes descritos. Para esto se utiliza el siguiente código mostrado en la figura 3.7.

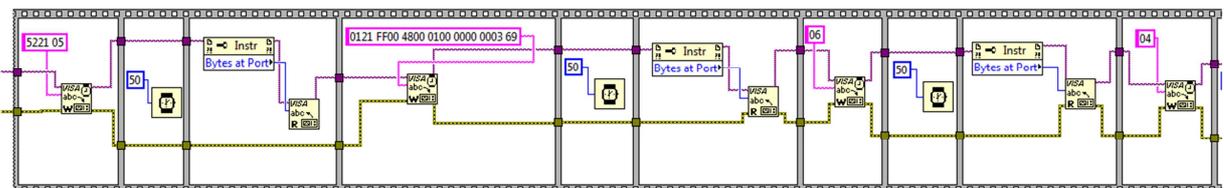


Figura. 3.7. Secuencia de Inicialización para el Controlador C500

Los bloques que permiten el envío de la secuencia de inicialización se describen brevemente a continuación:

VISA Write Function. Escribe una cadena de datos en el dispositivo o interfaz especificado anteriormente en la configuración. En la figura 3.8 se muestra el comando de escritura en el puerto Serial.



Figura. 3.8. Función de escritura

VISA Read Function. Lee el número especificado de bytes desde el dispositivo o interfaz y devuelve los datos en el búfer de lectura.

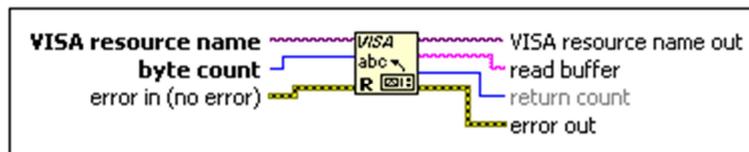


Figura. 3.9. Función de lectura

Luego que se envió y se acepto la secuencia de inicialización se procede a desactivar la ayuda del modo interactivo ya que esta autocompleta los comandos enviados y esto puede ser causa de error, además se encera una serie de parámetros para que el sistema funcione correctamente.

Entre los parámetros a inicializar es la orientación del Sistema de coordenadas y la longitud de la herramienta que este instalada. El siguiente código mostrado en la figura 3.10 se muestra lo dicho anteriormente.

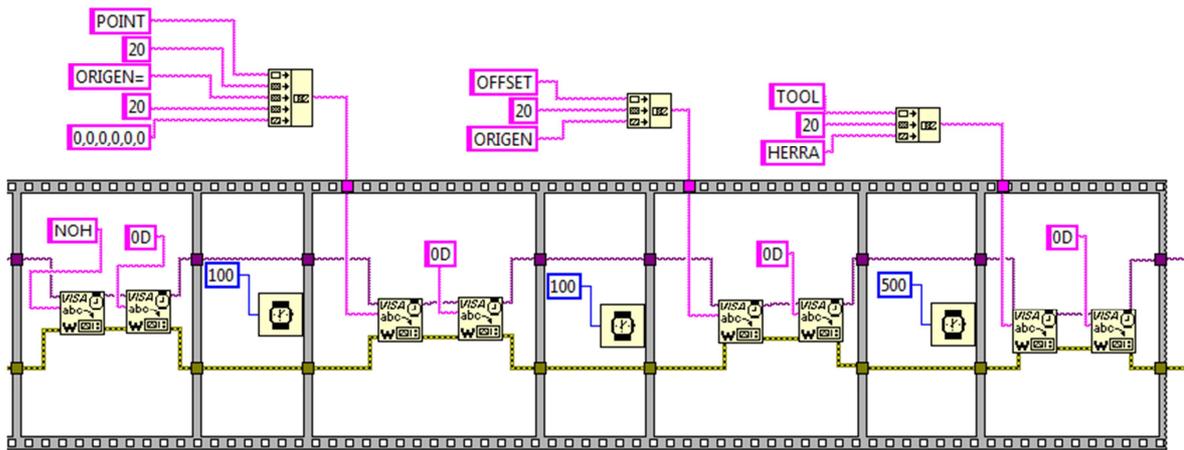


Figura. 3.10. Código de encieramiento

3.4.2 Condiciones de Funcionamiento

Para que se proceda al posicionamiento del robot, se debe superar una serie de condiciones, ya que de no ser así algunas de estas causarían errores en el funcionamiento del sistema.

Botón Mover. Este es el interruptor del sistema, para el funcionamiento debe estar activado.



Figura. 3.11. Botón Mover

Ciclo. Es un requisito que condiciona al sistema para que este realice una secuencia de movimiento a la vez, ya que si se envía muchos comandos, se almacenan en el buffer del controlador causando el error en la ejecución.

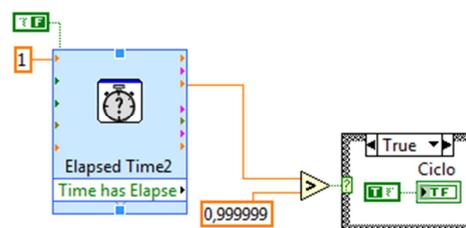


Figura. 3.12. Código de ciclo de funcionamiento

La figura 3.12 muestra el código que genera un tren de pulsos con características que se describen en la figura 3.13, el ancho del pulso solo permite el envío de una secuencia de movimiento a la vez.

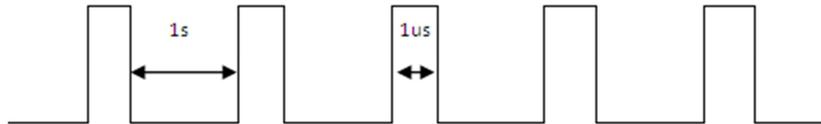


Figura. 3.13. Tren de pulsos del ciclo de funcionamiento

Movimiento terminado. Este código permite conocer si el robot se encuentra realizando algún movimiento anterior. Esto ayuda al flujo del programa, ya que solo si el robot termino de moverse permite el envío de instrucciones para un nuevo movimiento.

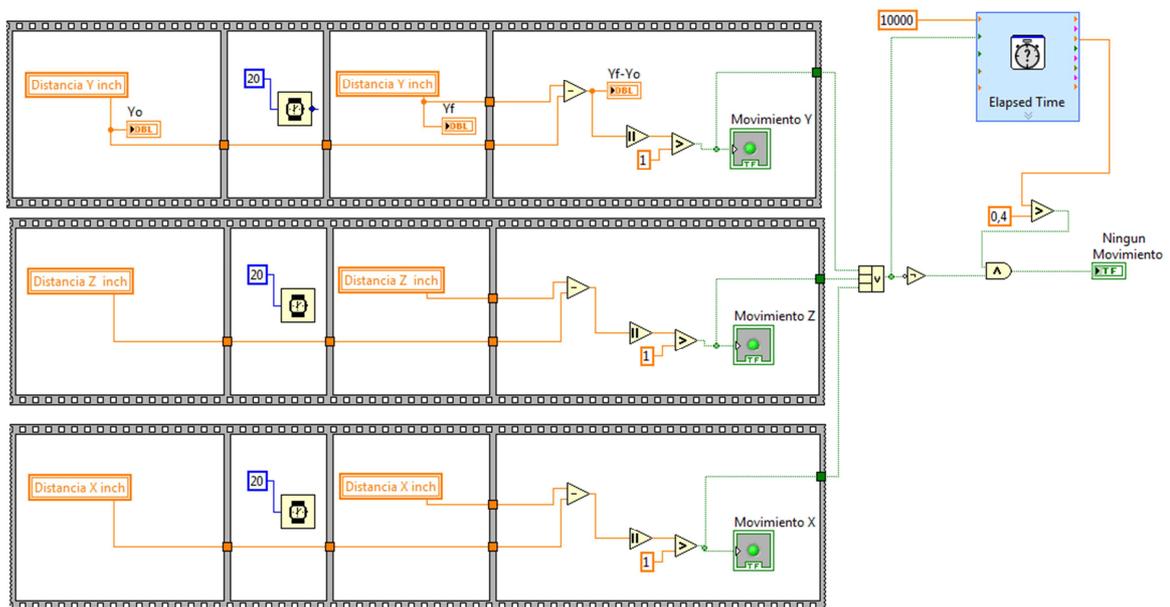


Figura. 3.14. Código de detección de movimiento

Para detectar el movimiento se realizó una comparación de la posición actual del objeto con una anterior en un tiempo específico. De esta manera si la posición del objeto es la misma refleja que el robot se encuentra detenido y viceversa.

Banda de SetPoint. Es un rango configurable que tiene el sistema, para variar la precisión de acercamiento al setpoint por parte del robot. Si el robot llega a posicionarse dentro de la banda no realiza más movimientos.

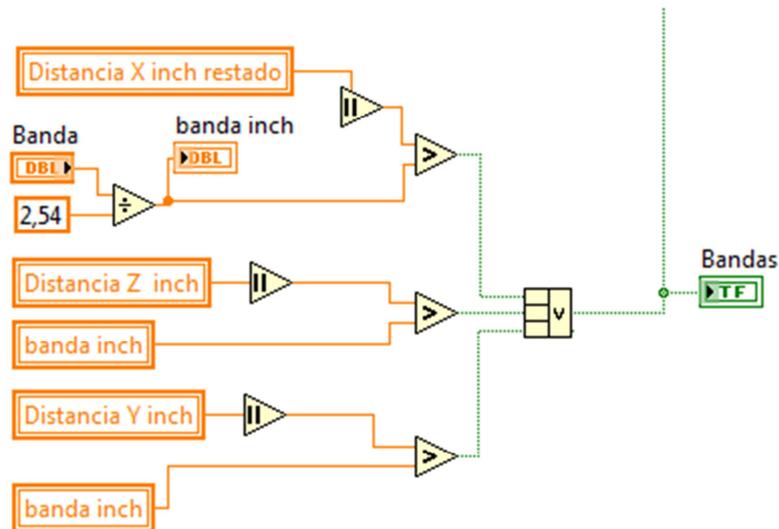


Figura. 3.15. Código de Banda de SetPoint

Enfocamiento Estéreo. Este condicionamiento es muy importante ya que si las dos cámaras no enfocan al mismo tiempo al objeto genera errores importantes en los datos de distancia enviados al controlador, para ello se realizó el siguiente código mostrado en la figura 3.16, que básicamente es una comparación con el resultado que nos arroja el comando de detección del objeto.

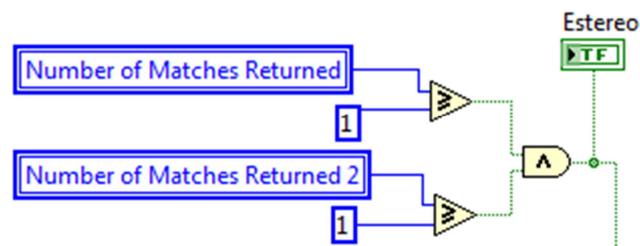


Figura. 3.16. Comprobación del enfocamiento estereo

3.4.3 Movimiento del Robot

Luego de superar el condicionamiento anterior el sistema ejecuta una secuencia con los datos obtenidos en el sistema de visión, la cual primero realiza un movimiento circular de la primera articulación del robot permitiendo llegar a la coordenada Y del *setpoint*. Para cumplir este primer desplazamiento se envía el comando JOINT con el ángulo calculado anteriormente como lo muestra la figura 3.17.

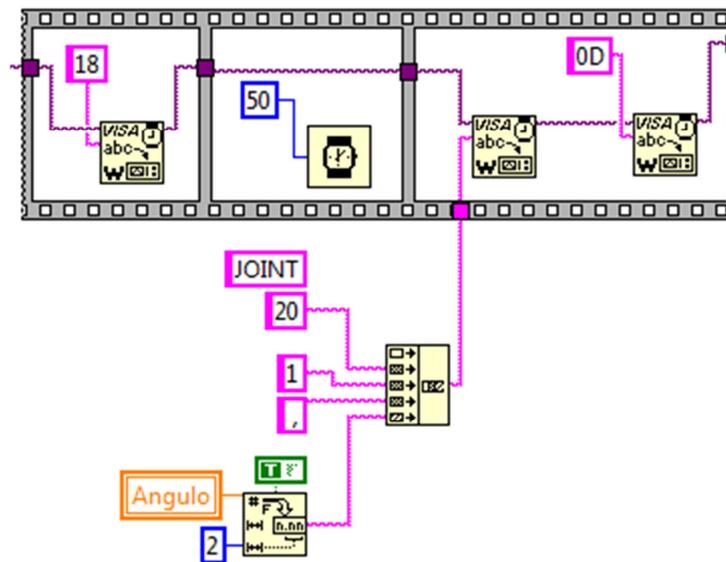


Figura. 3.17. Código de movimiento de la primera articulación

Posteriormente terminado el primer movimiento, se corrige la orientación del eje de coordenadas del robot. Una vez más se envía por medio del puerto serial con ayuda de las herramientas de la librería VISA los comandos POINT y OFFSET, como se muestra en la figura 3.18.

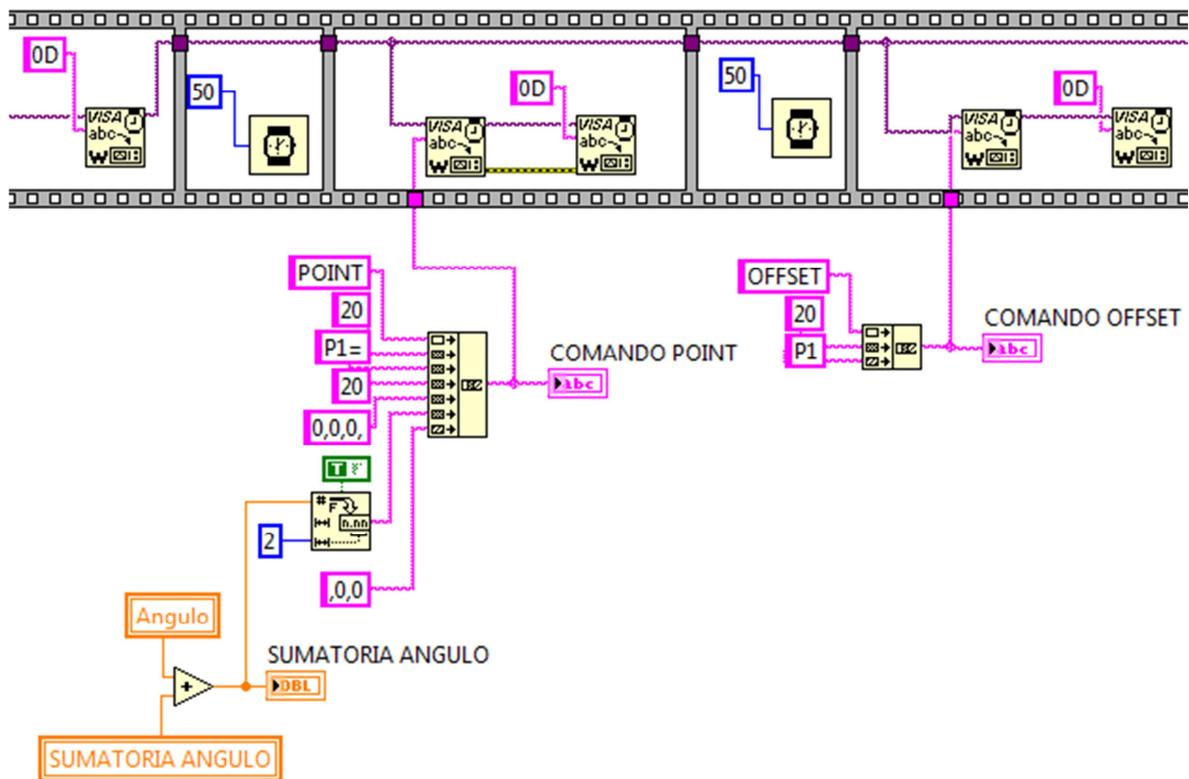


Figura. 3.18. Código de corrección de la orientación del eje de coordenadas

Para terminar la secuencia, se envía el comando JOG el cual permite que el robot se desplazarse en este caso en los ejes X y Z para alcanzar las coordenadas del SetPoint restantes. Para esta finalidad como se hizo anteriormente se manda el comando de forma serial como se muestra en la figura 3.19

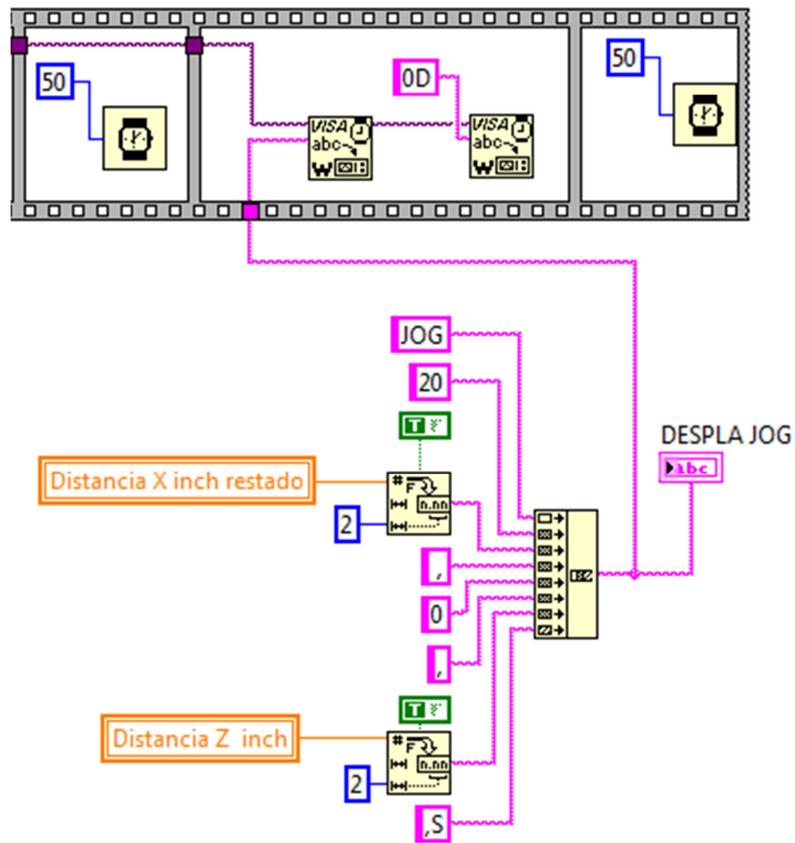


Figura. 3.19. Código de desplazamiento en los ejes X y Z

CAPÍTULO 4

PRUEBAS Y RESULTADOS

4.1 INTRODUCCIÓN

Uno de los propósitos del presente capítulo es describir las diferentes pruebas y métodos experimentales que se han realizado durante esta investigación. Se entenderá por pruebas a aquellos procesos que sirven para definir las características y limitaciones del sistema, por otro lado se entenderá como métodos a aquellos procesos que permiten estipular axiomas útiles dentro del desarrollo del sistema mediante el método experimental.

Otro de los propósitos de este capítulo es describir los resultados que se han obtenido al realizar las pruebas y métodos.

4.2 MÉTODOS

4.2.1 Obtención de la distancia.

A continuación se describe el método de aproximación por distancia y los resultados obtenidos

4.2.1.1 Descripción

Como ya se definió en el capítulo 2, la obtención de la distancia se basará en correspondencia de características entre las dos imágenes, en este caso el centro de un círculo. La disparidad es la diferencia entre las posiciones del plano imagen de la característica identificada.

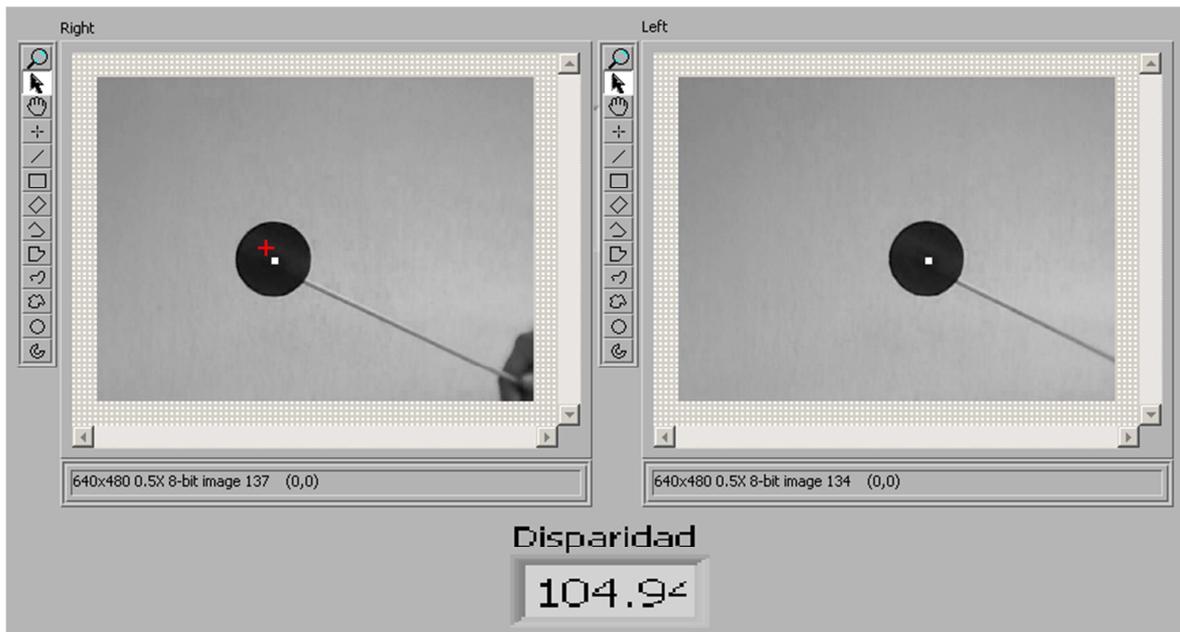


Figura. 4.1. Disparidad del objeto

La obtención de distancia en base a la técnica de visión estéreo se resume en la ecuación fundamental 1.2.

Donde:

- f es la distancia focal, parámetro intrínseco de la cámara, constante
- T es la distancia entre cámaras, constante
- d es la disparidad, variable.

Podemos observar que la distancia focal es un parámetro constante y que depende de la construcción de la cámara, en nuestro caso se desconoce este parámetro ya que las cámaras no lo especifican. La distancia entre cámaras es también una constante, se ha escogido la distancia más próxima entre cámaras, 50 mm., ya que permite mayor enfoque a cortas distancias. Entonces teóricamente se cumple la siguiente función relacional.

$$Z \propto \frac{1}{d} \quad (4.1)$$

O al agrupar términos constantes, $K = f \times T$, tenemos:

$$Z = \frac{K}{d} \quad (4.2)$$

En primer lugar se va a comprobar esta relación a través del recogimiento de muestras de disparidad a distancias conocidas. Los datos se recolectaron desde 130mm hasta 1000 mm, con intervalos de 20 mm. Se puede observar una fotografía del procedimiento experimental en el anexo C, figura toma de muestras, y las muestras en la tabla “muestras profundidad – disparidad”, del mismo anexo.

Graficando los datos del inverso de la disparidad en el eje de las abscisas y el de la distancia en el eje de las ordenadas, se obtiene la siguiente gráfica, ver figura 4.2.

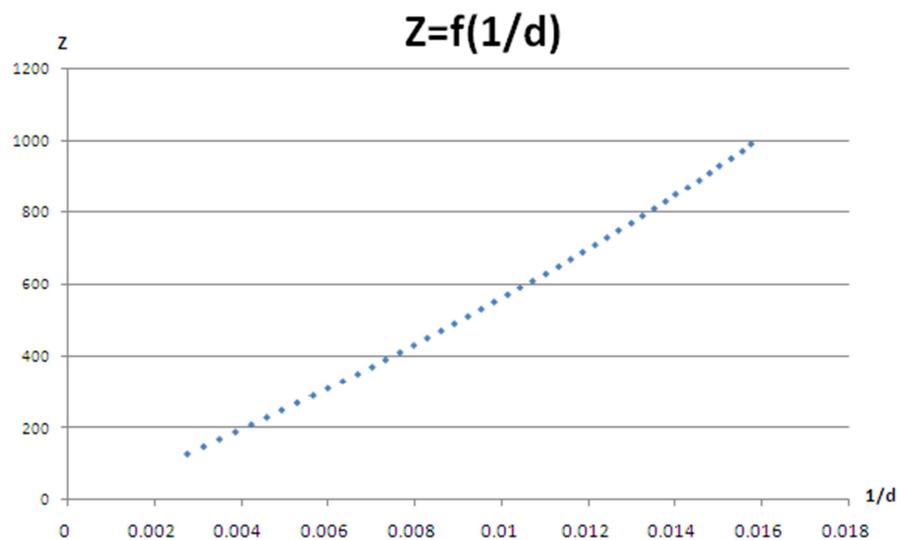


Figura. 4.2. Distancia vs inverso de la disparidad.

En la gráfica se puede apreciar claramente la proporcionalidad mencionada, se agrega una curva de tendencia lineal que cruce por el origen obteniéndose los siguientes resultados, representados en la figura 4.3.

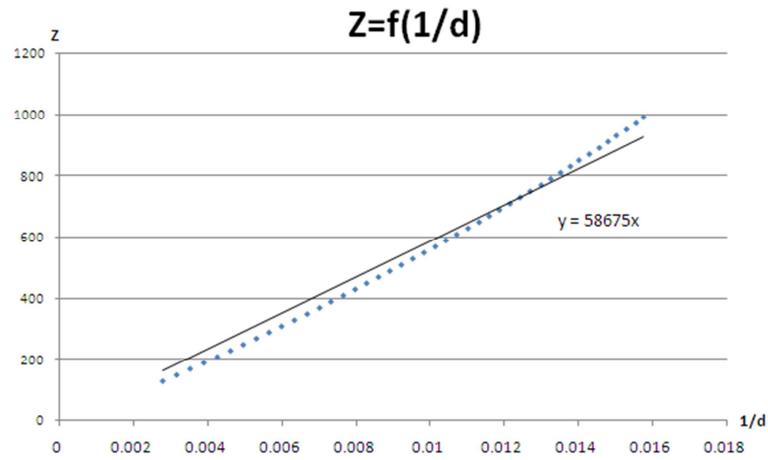


Figura. 4.3. Ajuste lineal cruce por el origen - profundidad

Ahora se analizará una curva de tendencia lineal que no cruce necesariamente por el origen, se observan los siguientes resultados, ver figura 4.4.

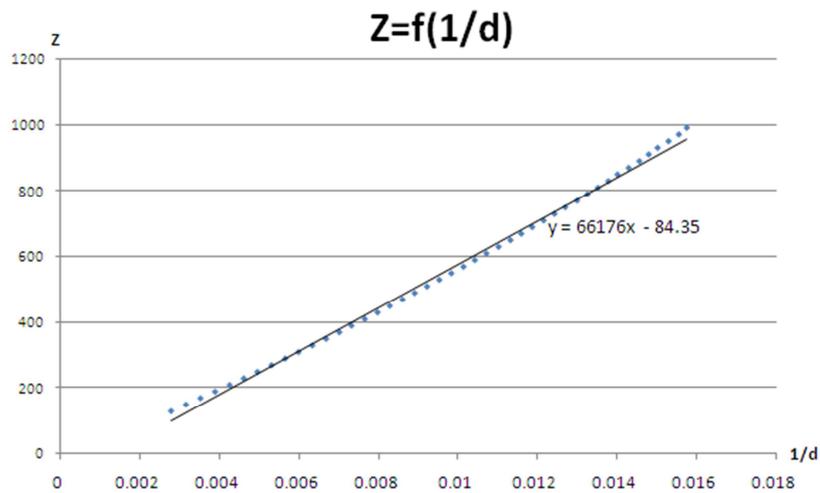


Figura. 4.4. Ajuste lineal - profundidad

Se ensaya también una curva polinómica de orden 2, ver figura 4.5.

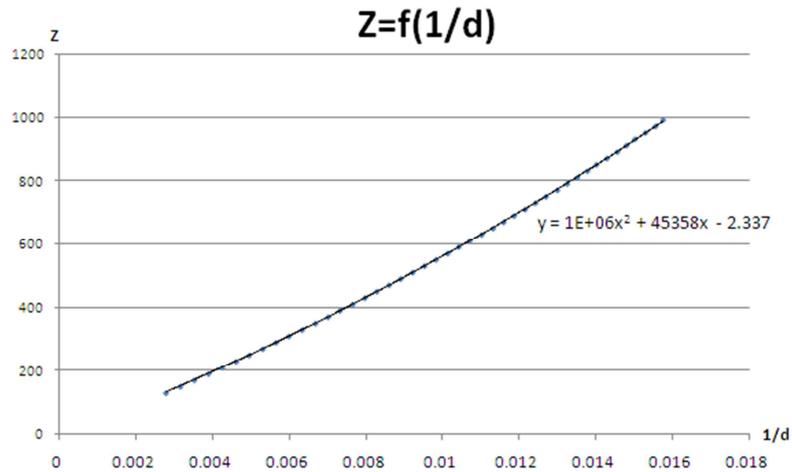


Figura. 4.5. Ajuste polinómico de grado 2 - profundidad

A continuación se evalúa teóricamente estas curvas y se obtienen los errores porcentuales y máximos, ver tabla “selección curva profundidad” del anexo C, siendo la curva polinómica de grado dos la que tiene menor error se la escoge e implementa.

4.2.1.2 Resultados

Una vez implementada la ecuación se evalúa su desempeño contrastando las distancias obtenidas con distancias reales. Además se determina el error porcentual, el error porcentual promedio y máximo, como se puede ver en la tabla “resultados curva de profundidad” del anexo C. En el siguiente gráfico se observa la diferencia entre la distancia real y la distancia estimada, ver figura 4.6.

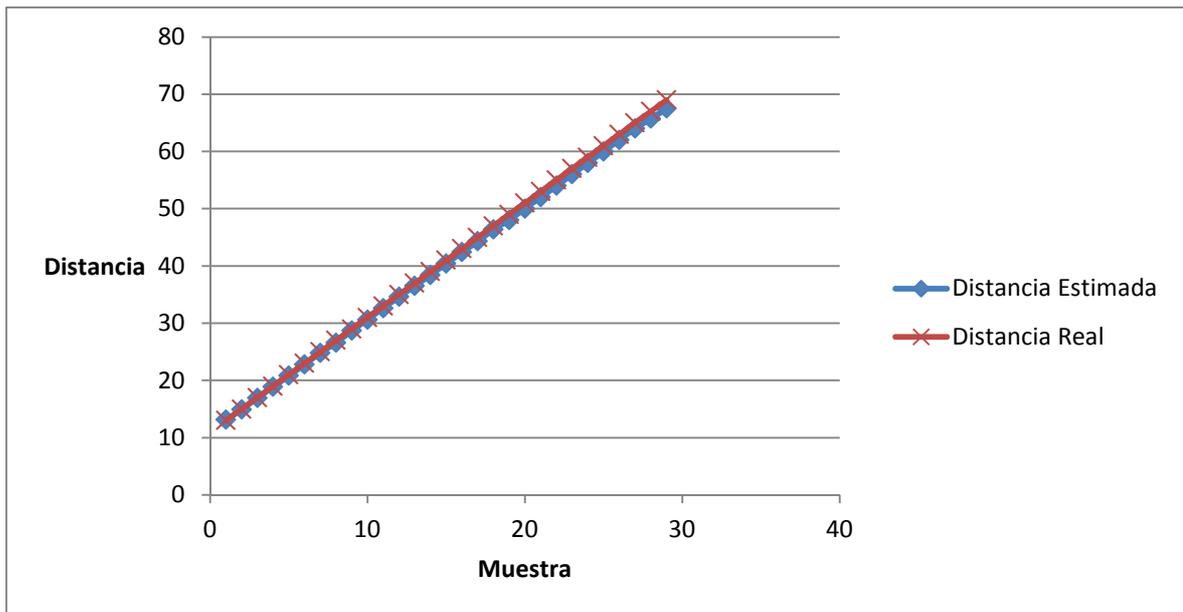


Figura. 4.6. Contraste valores de distancia

4.2.2 Obtención de la posición en el plano.

4.2.2.1 Descripción

Como ya se menciona la posición en el plano se realizará mediante el método de aproximación por distancia.

En primer lugar se va recoger muestras de un radio específico, 14.5 mm, en pixeles a distancias conocidas. Los datos se recolectaron desde 130mm hasta 1000 mm, con intervalos de 20 mm. Luego se divide el radio real entre el radio en pixeles y se obtiene VLP. Ver tabla “muestras VLP distancia” del anexo C.

A continuación se grafica VLP en función del inverso de la disparidad, se ensaya algunas curvas que se ajusten a los datos como se puede apreciar en las figura 4.7 y figura 4.8:

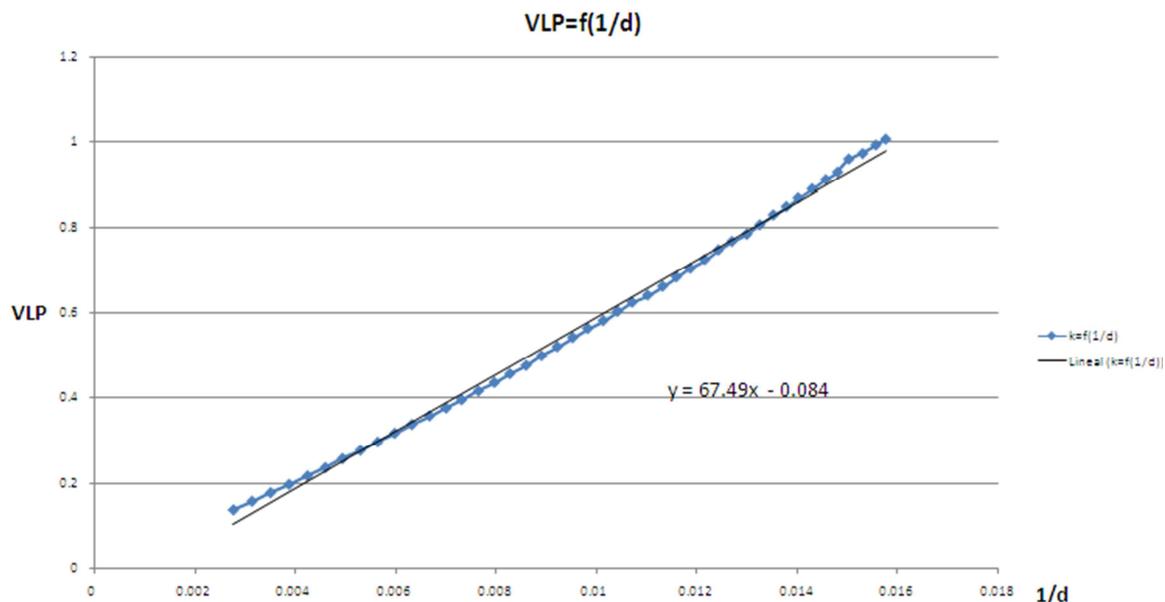


Figura. 4.7. Ajuste lineal - VLP

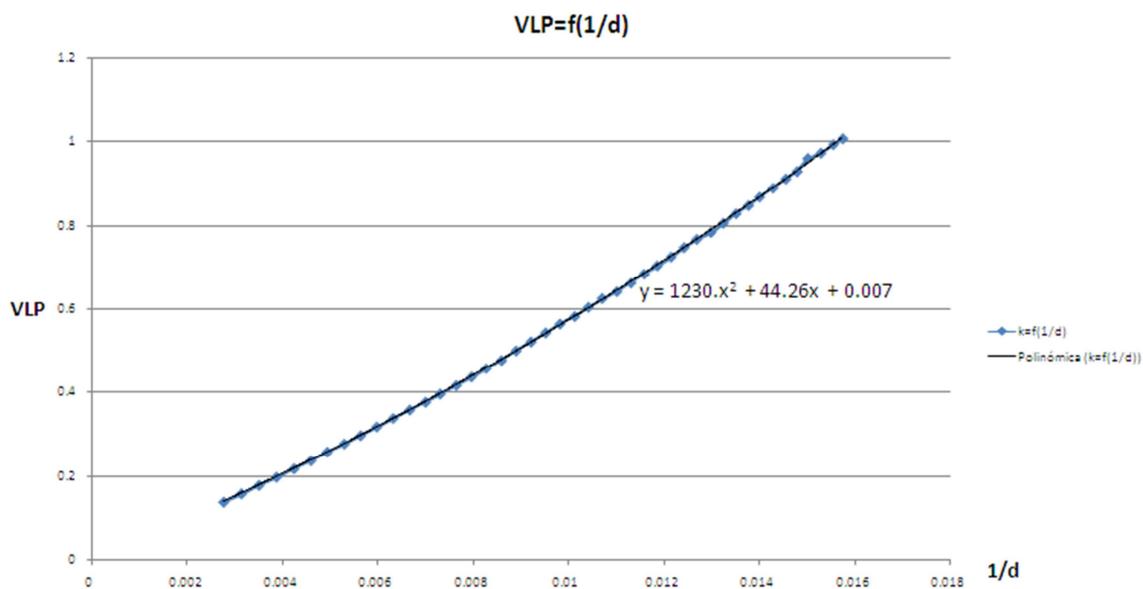


Figura. 4.8. Ajuste polinómico de grado 2 - VLP

A continuación se evalúa teóricamente estas curvas y se obtienen los errores porcentuales y máximos, ver tabla “selección curva VLP” del anexo C, siendo la curva polinómica de grado dos la que tiene menor error se la escoge e implementa.

4.2.2.2 Resultados

Una vez implementada la ecuación se evalúa su desempeño contrastando la constante VLP obtenida con VLP real. Además se determina el error porcentual, el error porcentual promedio y máximo, como se puede ver en la tabla “resultados curva VLP” del anexo C. En la figura 4.9 se observa la diferencia entre la distancia real y la distancia estimada.

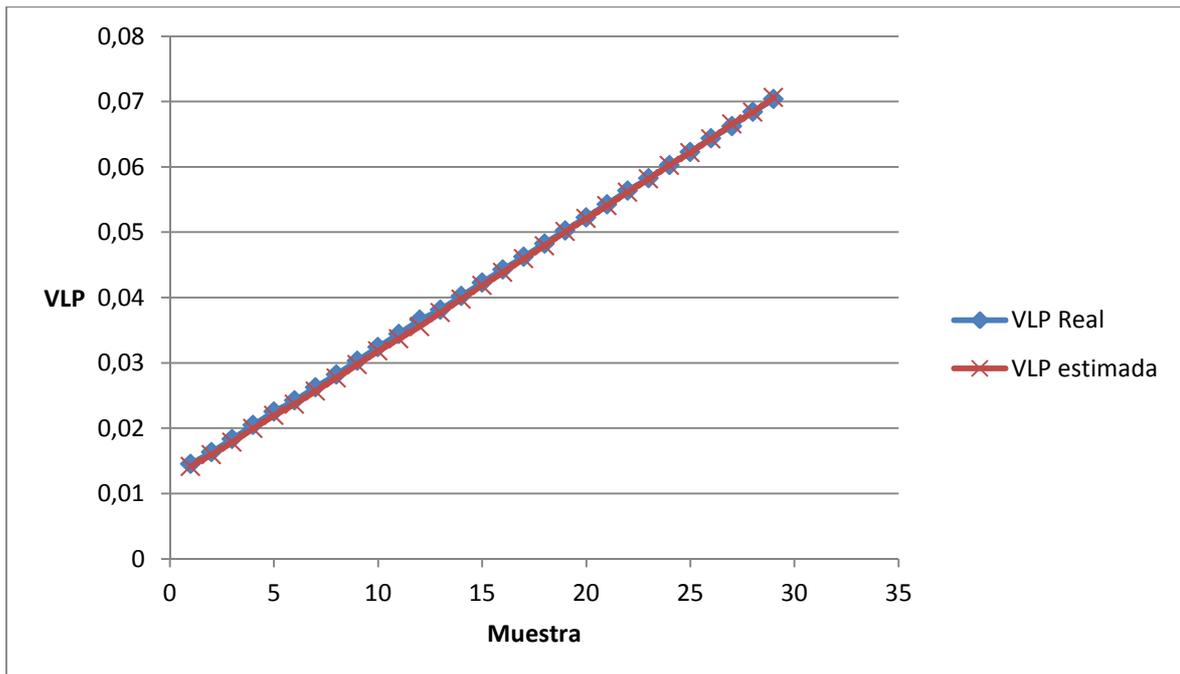


Figura. 4.9. Contraste valores de VLP

4.3 PRUEBAS

Se describen las diferentes pruebas a las que se ha sometido el sistema implementado.

4.3.1 Color del objeto



Figura. 4.10. Prueba de color del objeto

Con el objetivo de descubrir las limitaciones del sistema robótico con relación a la característica de color del objeto, se realizaron pruebas de funcionamiento. La prueba básicamente consistió en una secuencia de movimientos del manipulador robótico para cada objeto de diferente color, como se muestra en la figura 4.11. Esta experiencia intenta comprobar que si con los diferentes objetos de prueba el sistema robótico realiza la correspondencia y el posicionamiento correcto.



Figura. 4.11. Objetos de prueba por color

La prueba realizada genero los siguientes resultados:

Color	Detección(V ó F)	Posicionamiento (V ó F)
Rosado	V	V
Rojo	V	V
Celeste	V	V
Negro	V	V
Gris	V	V
Morado	V	V
Naranja	V	V
Azul	V	V
Verde Limón	V	V
Verde Oscuro	V	V
Amarillo	V	V

Tabla. 4.1. Resultados de la prueba de color

4.3.2 Deformación del objeto

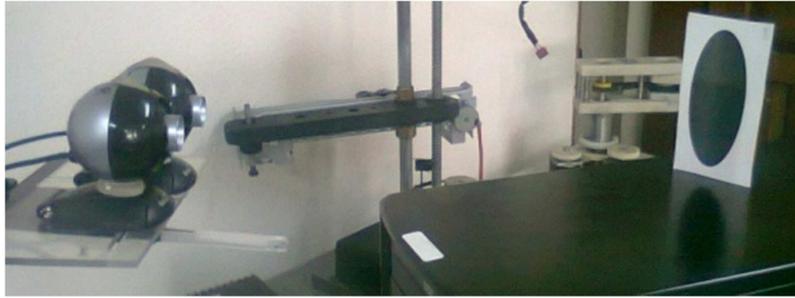


Figura. 4.12. Prueba de deformación del objeto

Otra característica que se debe comprobar es la deformación del objeto, para ello la experiencia se realizó con una serie de objetos circulares con deformaciones progresivas que tienden a ser elipses como se puede ver en la figura 4.13. La deformación crece a menudo que va aumentando el eje mayor de las elipses.

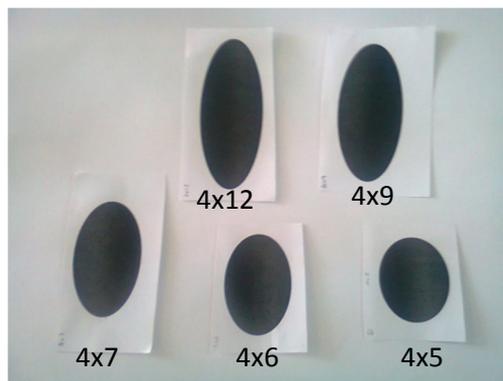


Figura. 4.13. Objetos de prueba por forma

Deformación	Detección(V ó F)	Posicionamiento (V ó F)
4X5	V	V
4X6	V	V
4X7	F	F
4X9	F	F
4X12	F	F

Tabla. 4.2. Resultados de la prueba de la deformación del objeto

Como se puede ver el sistema solo reconoce hasta a un nivel de deformación, ya que cuando el objeto tiende a parecer a otro objeto como en este caso una elipse el sistema lo descarta.

4.3.3 Tamaño del objeto



Figura. 4.14. Prueba de tamaño del objeto

Para conocer la respuesta que el sistema tiene hacia objetos de diferentes tamaños, de igual forma que las pruebas anteriores se generó movimientos repetitivos con diferentes objetos de prueba los cuales eran círculos con diferentes diámetros como se ve en la figura 4.15

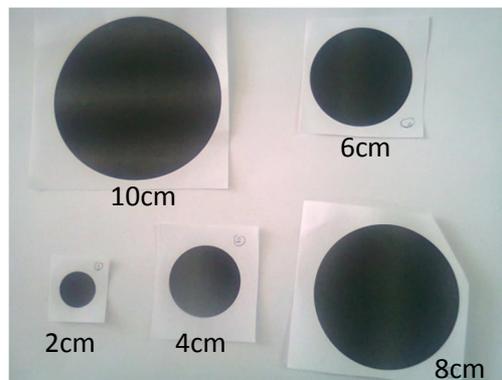


Figura. 4.15. Objetos de prueba por tamaño

La tabla 4.3 muestra que la prueba realizada con un círculo de 2 cm causó errores de funcionamiento del sistema robótico, este círculo no fue detectado al ser muy pequeño y por consecuencia el posicionamiento no se dio. Aunque esta variable de tamaño puede ser configurada en el sistema no se recomienda tenerla para círculos muy pequeños ya que puede causar confusiones con objetos del entorno. Además se puede decir que en la prueba con los círculos de 8 y 10 cm,

los objetos si fueron detectados pero el posicionamiento necesito de más movimientos ya que perdían el enfoque por ser muy grandes.

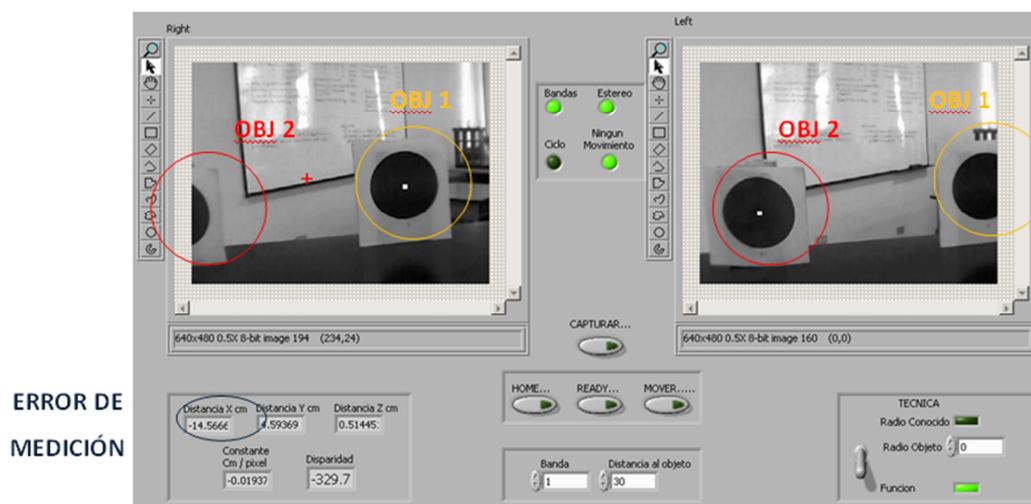
Tamaño (cm)	Detección(V ó F)	Posicionamiento (V ó F)
2	F	F
4	V	V
6	V	V
8	V	V
10	V	V

Tabla. 4.3. Resultados de la prueba de tamaño del objeto

4.3.4 Objetos múltiples

Para comprobar ciertas características del sistema estereoscópico implementado se llevo a cabo la siguiente experiencia, la cual consta de la presencia de dos objetos y sus consecuencias.

Como se puede ver en la figura 4.16, el sistema reconoce a los dos objetos como que si fueran correspondientes. La imagen de la cámara izquierda no es la misma que la imagen de la cámara derecha causando así error en la medición de la distancia y por consecuencia en el posicionamiento del manipulador. Para que el sistema estéreo funcione adecuadamente es completamente necesario que la imagen izquierda sea la misma o correspondiente a la derecha.



ERROR DE
MEDICIÓN

Figura. 4.16. Prueba de los Objetos múltiples

4.3.5 Volumen de trabajo

El volumen de trabajo de un robot se refiere únicamente al espacio dentro del cual puede desplazarse, ver la figura 4.17, en este caso es una limitación para el sistema ya que el robot solo puede posicionarse cuando el objeto este dentro de este.

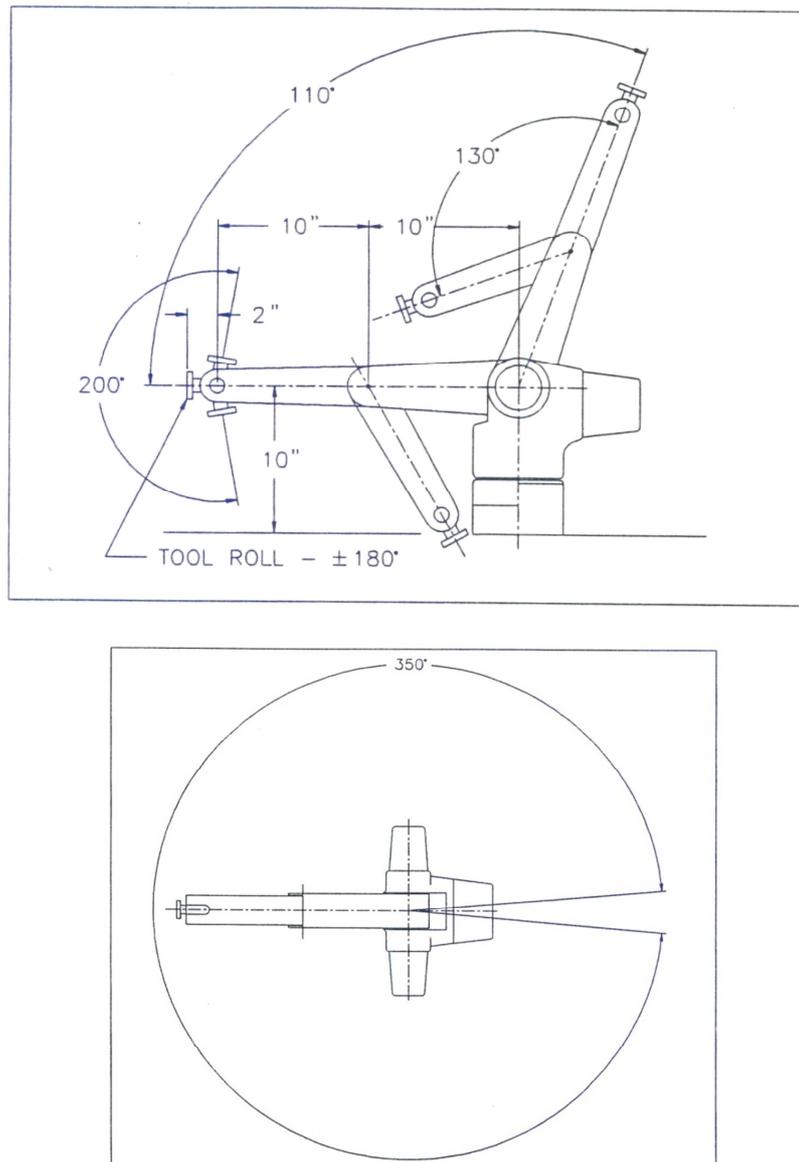


Figura. 4.17. Descripción del volumen de trabajo del manipulador

Se realizaron pruebas al sistema robótico para comprobar los posibles errores que podrían darse por motivo del posicionamiento fuera del volumen de trabajo. Estas pruebas básicamente consistieron en secuencias de movimiento

en las que se incluían posicionamientos dentro y fuera del espacio de trabajo del manipulador.

Los resultados de estos experimentos concluyeron, que si bien es cierto el controlador del robot da una señal de error cuando el objeto es inalcanzable, el sistema de posicionamiento no falla, sino que espera a una posición que si sea posible o alcanzable.

4.3.6 Velocidad del Manipulador

El controlador C500 posee varios niveles de velocidad de movimiento del manipulador, para comprobar si el funcionamiento del sistema es adecuado se realizaron posicionamientos a varias velocidades arrojando los siguientes resultados:

Velocidad (%)	Detección(V ó F)	Posicionamiento (V ó F)
10	V	F-V
15	V	V
20	V	V
25	V	V
30	V	V
50	V	V

Tabla. 4.4. Resultado de la prueba de velocidad

Aunque el sistema es configurable, las pruebas con los parámetros ya dispuestos generaron que el sistema no tuvo error en velocidades mayores a 15%, esto se debe a la sensibilidad del sensor de movimiento implementado en el sistema, que toma movimientos muy lentos como si el robot estuviera detenido.

4.3.7 Fondo del Objeto

Para reconocer las limitaciones del sistema de visión, se genero la prueba que pretende descubrir si el reconocimiento del objeto es posible con un fondo cualquiera o es necesario adecuar uno.

Para ello se colocó un objeto de prueba en primera ocasión con un fondo sin adecuación como se muestra en la figura 4.18. Lamentablemente el objeto no pudo ser detectado por el sistema ya que el objeto y el fondo no difieren en nivel de reconocimiento de píxeles, esto quiere decir no existe contraste entre los dos. Se puede mejorar la identificación de los píxeles que pertenecen al objeto mediante el proceso de segmentado, pero esto ocasionaría costo computacional causando la lentitud del sistema.



Figura. 4.18. Objeto con un fondo sin adecuación

Por otro lado, en el caso de detección en un ambiente con un fondo blanco el sistema no tuvo problemas en la detección del objeto como se demuestra en la figura 4.19.

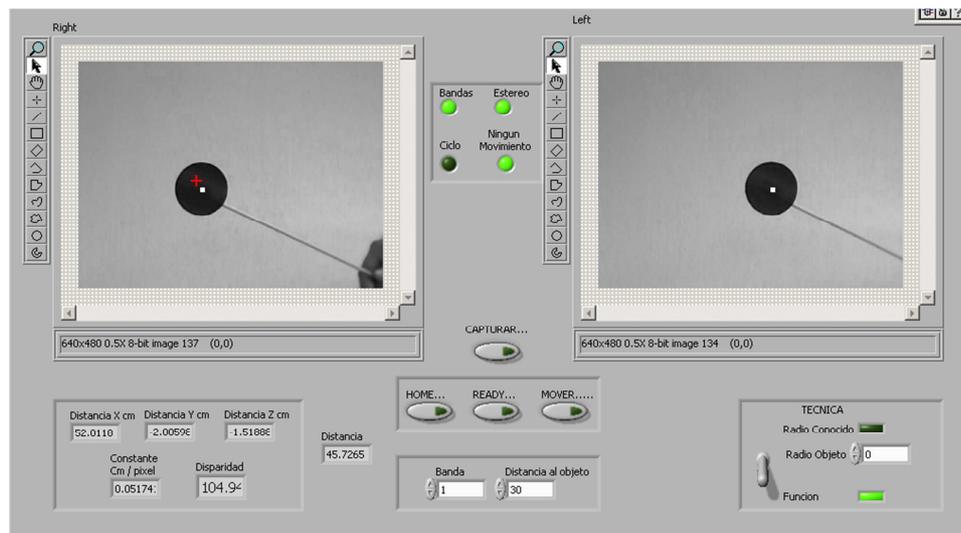


Figura. 4.19. Objeto con un fondo blanco

4.3.8 Niveles de iluminación

La iluminación es una de las características más importantes a tomar en cuenta en todo sistema de visión computacional, si bien es cierto como ya hemos dicho antes escogimos la detección de objetos por motivos de que este método no depende mucho de esta característica.

Para caracterizar los niveles de iluminación correctos para el funcionamiento del Sistema robótico, se procedió a una serie de pruebas con variación de la luminosidad. Se valió de un fotómetro Lamprobe Im-120, como se puede ver en la figura 4.20.



Figura. 4.20. Medición de la luminosidad

Variando la luminosidad del ambiente se tomaron mediciones en luxes, constatando que el sistema realice la función de reconocimiento del objeto, como se puede ver a continuación:



Figura. 4.21. Pruebas de iluminación del objeto

Esta serie de pruebas arrojaron los siguientes resultados:

Auto (lux)	Max (lux)	Min (lux)	Detección(V ó F)
0,2	0,35	0,08	F
0,51	0,77	0,34	F
1,65	1,69	1,37	F
3,78	3,98	3,77	V
20,5	20,5	20,5	V
30	30,1	30	V
65,1	65,9	64,9	V
112,3	112,8	111,5	V
125,1	125,6	125,1	V
140,8	142,2	139,2	V
230	231	230	V
251	253	251	V
581	558	610	V
1231	1301	1220	V
1866	1872	1624	V-F
2680	3010	2700	F

Tabla. 4.5. Resultados de la prueba de iluminación del Sistema

Luego de las pruebas se observa que el sistema tiene una banda considerable de iluminación para un funcionamiento adecuado. Esta banda va de 3,78 a 1231 luxes aproximadamente.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Al término de este proyecto se ha implementado un sistema de visión estereoscópica capaz de otorgar la posición espacial de un objeto en particular, además se ha desarrollado un algoritmo para el posicionamiento del manipulador robótico orientado a la interpretación de la información de posición del objeto.

Se ha estudiado los fundamentos de visión computacional y en especial la técnica de visión estereoscópica, su fundamento geométrico y los algoritmos computacionales generalmente usados.

Se ha elegido e implementado el algoritmo computacional más idóneo para la obtención de la ubicación espacial del objeto cumpliendo los requisitos de resolución, precisión, exactitud y error para la medición, enmarcada por el ambiente de trabajo y los dispositivos de captura.

Se ha analizado los diferentes tipos de opciones de interactividad que dispone el sistema robótico, determinándose que el modo interactivo es el más apto para el desarrollo del algoritmo de posicionamiento.

Se ha monitorizado y analizado el flujo de datos entre la interfaz de comunicaciones del controlador C500 y el software *Robcomm* actuando en modo interactivo, lo que ha permitido la emulación de este flujo.

De esta manera, la ejecución de comandos se realiza desde el entorno de programación escogido suprimiendo la necesidad de recurrir al software propietario *Robcomm*.

Se escogió al entorno de programación Labview 2009 para la implementación de todos los algoritmos necesarios; algoritmo de visión y posicionamiento. Ya que cuenta con las herramientas necesarias de visión computacional y comunicación serial.

Se determinó que la herramienta más competente para identificar al objeto de interés en las imágenes es el bloque de detección de formas definidas ya que el costo computacional y la dependencia de la iluminación es reducida en comparación a las otras herramientas de detección.

Mediante el método experimental se observó la tendencia lineal de la relación entre distancia y el inverso de disparidad, sin embargo para obtener menores medidas de error se utilizó una curva que se ajuste a los datos experimentales.

Para la determinación de VLP (variable longitud pixel), se han propuesto e implementado dos métodos llamados: De característica conocida y De aproximación por distancia.

El método de característica conocida es más exacto que el de aproximación por distancia pero necesita de la especificación del radio y la constancia de la orientación del objeto respecto a las cámaras. Lo que le resta versatilidad, además no involucra el conocimiento de la profundidad del cuerpo ubicado.

El método de aproximación por distancia es experimental, está basado en el de característica conocida y se vale de la profundidad del sistema estéreo. Por lo tanto el error es mayor que con el otro método. Sin embargo este error no es significativo para el posicionamiento del manipulador y la versatilidad se ve

incrementada al no ser necesario especificar el radio del cuerpo ni mantener la orientación del objeto con las cámaras.

El envío de la secuencia de movimiento se controló mediante las condiciones de funcionamiento (ciclo, movimiento terminado, banda y enfoque estéreo) evitando errores de sobrecarga del buffer del controlador, toma de datos de ubicación errónea y movimientos innecesarios del manipulador.

Se implementó un algoritmo de movimiento que resuelve la problemática del posicionamiento sucesivo del manipulador así como la orientación del eje coordinado minimizando la cantidad de comandos y procesamiento.

Se realizó diferentes pruebas para comprobar el funcionamiento del sistema, los resultados se recogen en la tabla 5.1:

Prueba Realizada	Resultados
Color	Ningún color de los probados ocasiono mal funcionamiento
Deformación	Elipses de 4x5 y 4x6 fueron detectadas, las de mayor deformación no se registraron
Tamaño	Aunque este parámetro puede ser configurado, el sistema con las configuraciones ya dispuestas, solo respondió a mayores de 4 cm a distancias cercanas
Objetos múltiples	La presencia de objetos circulares múltiples en la escena puede ocasionar fallas en la medición de distancias
Volumen de trabajo	Si bien es cierto el controlador del robot da una señal de error cuando el objeto es inalcanzable, el sistema de posicionamiento no falla, sino que espera a una posición que si sea posible o alcanzable.
Fondo del objeto	El fondo tiene q contrastar uniformemente con la circunferencia a ser detectada, caso contrario el objeto no será registrado
Niveles de iluminación	El sistema tiene una banda considerable de iluminación para un funcionamiento adecuado. Esta banda va de 3,78 a 1231 luxes aproximadamente.
Velocidad del manipulador	Aunque el sistema es configurable, las pruebas con los parámetros ya dispuestos generaron que el sistema no tuvo error en velocidades mayores a 15%

Tabla. 5.1. Resumen de resultados

5.2 RECOMENDACIONES

Se recomienda que en base al algoritmo de comunicación desarrollado, que mejora notablemente la facilidad de programación del robot, se desarrollen nuevas investigaciones como, operación remota inalámbrica del manipulador, rescate de información desde base de datos, uso de dispositivos incompatibles en el software *Robcomm* entre otras.

Se propone dar otras aplicaciones al sistema de visión ya que la ubicación espacial que realiza es fiable y altamente utilizable en sistemas robóticos.

Los algoritmos de visión desarrollados podrían utilizarse con dispositivos de captura de mejores características sin que esto implique un cambio significativo en el sistema.

Utilizando los dispositivos de captura se sugiere implementar nuevos algoritmos de visión estereoscópica que satisfagan otras necesidades como la percepción de la profundidad de todos los objetos de una escena (mapa de disparidad), modelamiento 3D u otras.

Se recomienda incrementar la versatilidad en cuanto a la detección del objeto de interés, sin embargo se advierte que esto implicaría un mejoramiento de la tecnología a utilizarse por el consumo de recursos que se ocasiona.

Debido a la importancia que ha adquirido la rama de visión computacional dentro de la electrónica se propone realizar investigaciones dentro de esta rama especialmente en el área de procesamiento de imágenes, obtención de características y detección de objetos.

BIBLIOGRAFÍA

- [1] SUCAR, Luis, *Procesamiento de imágenes y visión computacional*, Tecnológico de Monterrey, Cuernavaca.
- [2] CRS, *CRS A255 Robot installation and operation manual*, CRS, Ontario, 1993.
- [3] <http://omarsanchez.net/vision.aspx>, Sistemas de visión
- [4] MURAT, Tekalp, *Digital video processing*, Prentice Hall, New Jersey, 1995
- [5] VAINA, L., *From the retina to the neocortex: selected papers of David Marr*, Birkhauser, Boston, 1990.
- [6] MCNOUGHTON, B., *Commentary on simple memory: a theory of the archicortex*. Birkhauser, Boston, 1990.
- [7] LEW, Michael, *Principles of visual information retrieval*, Springer, 2001.
- [8] LECUMBERRY, Federico, *Cálculo de disparidad y segmentación de objetos en secuencias de video*, Universidad de la República, Montevideo, 2005.
- [9] HERNANDEZ, A., "Sistema clasificador de imágenes en web." [Online] Mayo 2005. http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/hernandez_s_a/portada.html.
- [10] DEL BIMBO, Alberto, *Visual information retrieval*, Morgan Kaufmann, 1999.

ÍNDICE DE FIGURAS

CAPÍTULO 1: MARCO TEÓRICO

Figura. 1.1. Esquema general del procesamiento de imágenes.	2
Figura. 1.2. Esquema general de visión por computadora.	3
Figura. 1.3. Ejemplo procesamiento de imágenes.	4
Figura. 1.4. Ejemplo sistema de visión.	4
Figura. 1.5. Imagen 2D representada en 3D.	6
Figura. 1.6. Modelo RGB.	8
Figura. 1.7. Modelo CMY 9	9
Figura. 1.8. Modelos CMYK y RGB.	9
Figura. 1.9. Modelo HSV 10	10
Figura. 1.10. Modelo HLS 10	10
Figura. 1.11. Espectro Electromagnético 15	15
Figura. 1.12. Arquitectura Básica Sistema de Visión.	16
Figura. 1.13. Ejemplo algoritmo estéreo básico 18	18
Figura. 1.14. Geometría de ejes paralelos 20	20
Figura. 1.15. Triangulación y Correspondencia.	21
Figura. 1.16. Calculo de la profundidad.	22
Figura. 1.17. Restricción de orden 24	24
Figura. 1.18. Costo Computacional.	25
Figura. 1.19. Brazo articulado TRA-14-203.	28
Figura. 1.20. Sistema Coordinado del Mundo.	30

Figura. 1.21. Sistema Coordenado Cilíndrico.....	31
Figura. 1.22. Sistema Coordenado de las articulaciones	32
Figura. 1.23. Panel Frontal Controlador C500.....	33
Figura. 1.24. Vista del Panel posterior del Controlador C500.	35

CAPÍTULO 2: SISTEMA DE VISIÓN ESTEREOSCÓPICA

Figura. 2.1. Ejes de ubicación de objeto	38
Figura. 2.2. Coordenadas en pixeles de un objeto	44
Figura. 2.3. Variación de pixeles	44
Figura. 2.4. Ejemplificación de la variación de VLP.....	46
Figura. 2.5. Cambio de características por propiedades físicas.....	47
Figura. 2.6. Cambio de características por alineación.....	47
Figura. 2.7. Arquitectura del sistema de Visión	50
Figura. 2.8. Programa para la adquisición de imágenes	54
Figura. 2.9. Bloque Imaq Create	55
Figura. 2.10. Bloque Imaqdx Open Camera.....	55
Figura. 2.11. Bloque Imaqdx Configure Grab.....	56
Figura. 2.12. Bloque Imaqdx Grab	57
Figura. 2.13. Flujo para la adquisición de alto nivel.....	57
Figura. 2.14. Extracción del plano de color de las imágenes	58
Figura. 2.15. Detección de círculos	61
Figura. 2.16. Extracción de información relevante	62
Figura. 2.17. Cálculo de la disparidad y la distancia	62
Figura. 2.18. Descripción de la ubicación de setpoint	63
Figura. 2.19. Técnica del Radio conocido	64
Figura. 2.20. Técnica de la Función experimental.....	64

Figura. 2.21. Transformación de las distancias en pixeles a centímetros	64
---	----

CAPÍTULO 3: POSICIONAMIENTO DEL BRAZO ROBÓTICO

Figura. 3.1. Ventana del Programa Monitor del Puerto Serial	67
Figura. 3.2. Escenario inicial	72
Figura. 3.3. Primer movimiento.	73
Figura. 3.4. Movimiento inicial	74
Figura. 3.5. Modificación del sistema con YAW	75
Figura. 3.6. Configuración e Inicialización del puerto Serial.....	77
Figura. 3.7. Secuencia de Inicialización para el Controlador C500	78
Figura. 3.8. Función de escritura.....	79
Figura. 3.9. Función de lectura.....	79
Figura. 3.10. Código de enceramamiento	80
Figura. 3.11. Botón Mover	80
Figura. 3.12. Código de ciclo de funcionamiento	80
Figura. 3.13. Tren de pulsos del ciclo de funcionamiento	81
Figura. 3.14. Código de detección de movimiento	81
Figura. 3.15. Código de Banda de SetPoint.....	82
Figura. 3.16. Comprobación del enfocamiento estéreo.....	82
Figura. 3.17. Código de movimiento de la primera articulación.....	83
Figura. 3.18. Código de corrección de la orientación del eje de coordenadas	84
Figura. 3.19. Código de desplazamiento en los ejes X y Z	85

CAPÍTULO 4: PRUEBAS Y RESULTADOS

Figura. 4.1. Disparidad del objeto	87
Figura. 4.2. Distancia vs inverso de la disparidad.	88
Figura. 4.3. Ajuste lineal cruce por el origen - profundidad	89
Figura. 4.4. Ajuste lineal - profundidad.....	89
Figura. 4.5. Ajuste polinómico de grado 2 - profundidad	90
Figura. 4.6. Contraste valores de distancia	91
Figura. 4.7. Ajuste lineal - VLP	92
Figura. 4.8. Ajuste polinómico de grado 2 - VLP	92
Figura. 4.9. Contraste valores de VLP	93
Figura. 4.10. Prueba de color del objeto	93
Figura. 4.11. Objetos de prueba por color	94
Figura. 4.12. Prueba de deformación del objeto	95
Figura. 4.13. Objetos de prueba por forma	95
Figura. 4.14. Prueba de tamaño del objeto	96
Figura. 4.15. Objetos de prueba por tamaño.....	96
Figura. 4.16. Prueba de los Objetos múltiples.....	97
Figura. 4.17. Descripción del volumen de trabajo del manipulador	98
Figura. 4.18. Objeto con un fondo sin adecuación	100
Figura. 4.19. Objeto con un fondo blanco	101
Figura. 4.20. Medición de la luminosidad	102
Figura. 4.21. Pruebas de iluminación del objeto	102

ÍNDICE DE TABLAS

CAPÍTULO 1: MARCO TEÓRICO

Tabla. 1.1. FPS sistema comercial.....	28
--	----

CAPÍTULO 2: SISTEMA DE VISIÓN ESTEREOSCÓPICA

Tabla. 2.1. Dispositivos de la arquitectura del sistema.....	49
Tabla. 2.2. Especificaciones del Dispositivo de Captura	51
Tabla. 2.3. Características CPU.....	51
Tabla. 2.4. Características de video	52
Tabla. 2.5. Nombramiento de las Cámaras.....	56
Tabla. 2.6. Planos de color.....	58

CAPÍTULO 3: POSICIONAMIENTO DEL BRAZO ROBÓTICO

Tabla. 3.1. Códigos de control ASCII	69
Tabla. 3.2. Descripción del contenido de la Cabecera	69

CAPÍTULO 4: PRUEBAS Y RESULTADOS

Tabla. 4.1. Resultados de la prueba de color.....	94
Tabla. 4.2. Resultados de la prueba de la deformación del objeto.....	95
Tabla. 4.3. Resultados de la prueba de tamaño del objeto	97

Tabla. 4.4. Resultado de la prueba de velocidad	99
Tabla. 4.5. Resultados de la prueba de iluminación del Sistema	103

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

Tabla. 5.1. Resumen de resultados	106
---	-----