

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA
AUTOMATIZACIÓN Y CONTROL

PROYECTO DE GRADO PARA LA OBTENCIÓN DEL
TÍTULO DE INGENIERÍA

ALGORITMO PARA LA UBICACIÓN DE OBJETOS
UTILIZANDO UNA RED DE VISIÓN ESTEREOSCÓPICA

DANIEL SÁENZ DE VITERI CABEZAS

Sangolquí – Ecuador
2008

CERTIFICACIÓN

Certificamos que el presente proyecto de grado fue realizado en su totalidad por el Sr. Daniel Sáenz de Viteri bajo nuestra dirección.

Ing. Julio Larco
DIRECTOR

ing. Alex Chacón
CODIRECTOR

Dedicatoria:
A papá y mama

Agradecimientos:

Al ingeniero Julio Larco y a mi tía Alicia de Ramírez
por la ayuda especial dedicada a la
culminación de este proyecto de tesis.

PROLOGO

Una red de visión está formada por un conjunto de elementos como: pares de cámaras dispositivos para análisis de longitud de onda, cámaras infrarrojas, de disparo temporizado, entre otras, que nos permiten tener una percepción de imágenes de algo que se desea. Por ejemplo tenemos los circuitos de CCTV (Circuito Cerrado de Televisión), cámaras estereoscópicas, ciertos microscopios.

En una red de visión, las estructuras más utilizados son las CCTV's para sistemas de seguridad, el par de cámaras infrarrojas-ultravioletas para análisis de maquinaria, cámaras montadas en brazos mecánicos para televisión y una de las estructuras que más se está utilizando es la visión estereoscópica debido a la gran cantidad de análisis que nos permite realizar del ambiente al cual las cámaras están enfocando.

La visión estereoscópica es un método de procesamiento de imágenes que utiliza dos cámaras; para extraer información útil, para realizar una determinada operación de monitoreo o control; como ejemplos de visión estereoscópica tenemos la visión animal que con un par de ojos dispuestos frontalmente le permite al animal la percepción de profundidad de los objetos, otro ejemplo de visión estereoscópica son las imágenes tomadas desde aviones que usan los geólogos para el estudio del suelo; se realizan dos tomas fotográficas separadas con una distancia determinada de tal manera que den una percepción de profundidad.

Procesos como la detección de fallas en la manufactura industrializada son controlados por visión, un sistema mixto de láser y cámara de visión permite ubicar objetos en el espacio, sistemas de seguridad son desde hace mucho tiempo implementados con sistemas CCTV, sin embargo ahora ya se dispone de sistemas automáticos que pueden incluso detectar cuando una persona está en problemas, como tumbado en el piso, e informar inmediatamente sin la ayuda de ninguna supervisión humana.

Existen algunas técnicas para la creación de la visión estereoscópica artificial, entre las principales tenemos imágenes estereoscópicas estáticas como las que utilizan los geólogos; visión estereoscópica de imagen única polarizada, esta es la que veremos comúnmente utilizarse en las salas de cine en donde las dos imágenes se encuentran en una misma pantalla y se las despolariza con gafas de lente pigmentado.

Entre las aplicaciones más útiles están: la eliminación de trazas, ruido, entrelazados fibrosos en las imágenes, pero la principal utilidad es la de la percepción de la profundidad, esta técnica es muy utilizada en la actualidad por sistemas autónomos como robots, sistemas de guía, etc. Esta percepción de profundidad se utiliza para crear principalmente mapas espaciales que a un robot le permitiría saber su posición actual y a la que tiene que llegar, el uso de estos mapas permiten identificar obstáculos y nuevas rutas de acceso a su destino.

Este proyecto está destinado a encontrar un algoritmo que permita que estas capacidades puedan ser realizadas en sistemas más complejos con un costo bajo de procesamiento, considerable precisión y error mínimo.

ÍNDICE DE CONTENIDO

Capítulo 1	1
1. INTRODUCCIÓN	1
1.1. Descripción de proyecto, alcances	2
1.2. Conceptos principales	3
Capítulo 2	5
2. EQUIPOS UTILIZADOS EN VISIÓN ARTIFICIAL	5
2.1. Cámaras	6
2.2. Protocolos	12
2.3. Redes de Visión	17
Capítulo 3	21
3. ESTEREOSCOPIA	21
3.1. Definición	21
3.2. Razones para el uso de estereoscopia	25
3.3. Posibles soluciones con algoritmos	26
Capítulo 4	29
4. DISEÑO DEL ALGORITMO	29
4.1. Funciones principales	29
4.2. Estructura Switch On/Off Y Casos De Excepción:	31
4.3. EJEMPLO DEMOSTRATIVO:	45
Capítulo 5	49
5. DESARROLLO DEL ALGORITMO	49
5.1. El lenguaje	49
5.2. Las cámaras	50
5.3. El algoritmo	51
5.4. Introducción al lenguaje:	52
5.5. El programa.	53
Capítulo 6	67
6. Pruebas y resultados	67
6.1. Comparaciones con otros métodos:	79
Capítulo 7	81
7. Conclusiones y Recomendaciones	81
7.1. Conclusiones	81
7.2. Recomendaciones	81
Referencias Bibliográficas	83

INCIDE DE FIGURAS

Capítulo 1	1
1.1. Descripción de proyecto, alcances	2
1.2. Conceptos principales	3
Capítulo 2	5
Figura 2.1: Bloque de procesamiento de imágenes	5
Figura: 2.2 Planos de enfoque	6
Figura: 2.3 Desenfoque de objetos	7
Figura: 2.4 Geometría de cámaras de un solo lente	8
Figura: 2.5 Efecto de enfoque	9
Figura: 2.6 Difracción de la luz en cámaras	11
Figura: 2.7 Imágenes por canal	17
Figura: 2.8 Topologías de redes	19
Capítulo 3	21
Figura: 3.1 Planos de observación con giro de cámara	21
Figura: 3.2 Planos de observación con giro de cámara	23
Figura: 3.3 Ángulo sólido de visión	1
Figura: 3.4 Redondeo de píxeles en la captura de imágenes	24
Figura: 3.5 Centro de masas en objetos digitalizados	25
Figura: 3.5 Algoritmo de posicionamiento de cámaras	27
Figura: 3.6 Algoritmo de ubicación sin posicionamiento de cámaras	28
Capítulo 4	29
Figura: 4.1 Triangulación simple sin ángulos obtusos	32
Figura: 4.2 Triangulación con ángulos obtusos	33
Figura: 4.3 Ploteo de puntos Distancia Focal vs Ángulo de Visión	37
Figura: 4.4 Recta de ajuste sobre puntos ploteados	38
Figura: 4.5 Error residual tabulado	39
Figura: 4.6 Ángulos dentro de la triangulación simple	42
Figura: 4.7 Distancias para cálculo de ángulos en la imagen	1
Figura: 4.8 Simulación de imágenes para un ejemplo de ángulos agudos	45
Capítulo 5	49
Figura 5.1 Diagrama de Flujo del Algoritmo	51
Figura: 5.1 Programación del algoritmo base en LabView	54
Figura 5.2: Bloque 1 Reserva de memorias y configuración de cámaras	55
Figura 5.4 Ventana de selección de ROI	57
Figura 5.5: Bloque 3 Sección de búsqueda de tramas	58
Figura 5.6: Bloque 4 timer de operaciones	59
Figura 5.7: Bloque 6 Presentación de resultados	59
Figura 5.8: Bloque 7 ActiveX para representación en figura 3D	60
Figura 5.9: Bloque 8 Condiciones de fin de programa	60
Figura: 5.10 Algoritmo programado en LabView	62
Figura 5.11: Switch on off para identificar triángulos agudos u obtusos	63
Figura 5.12: Determinación de ángulos mayores o menores a 90°	64
Figura 5.11: Separación de los datos y conversión a coordenadas rectangulares	66
Capítulo 6	67
Figura: 6.1 Error sobre la distancia	67
Figura 6.2: Prueba de escritorio	71
Figura 6.3: Dispersión de puntos medidos vs encontrados	72
Figura 6.3: Extrapolación de errores de datos medidos	78
Figura 6.4: Dispersión de las distancias vs error medio	79

INDICE DE TABLAS

Capítulo 3	21
Tabla:4.1 Distancia Focal vs Angulo de Visión	36
Tabla: 4.2 Tabla residual	40
Tabla: 4.3 Historial Chi Cuadrado	41
Capítulo 6	67
6. Pruebas y resultados	67
Tabla 6.2: Valores medidos por el algoritmo y promedios	71
Tabla 6.3 Tabulación de tiempos en 10 búsquedas de 10s.	76
Tabla 7.1 Distancia vs Error cuadrático medio	78

GLOSARIO

Visión (artificial)

Es la adquisición continua de imágenes por medio de un dispositivo. Esta se la hace a razón de una tasa de adquisición de imágenes con el objetivo de programar un computador para que "entienda" una escena o las características de una imagen.

Tasa de adquisición de imagen

Es la cantidad de cuadros por segundos o fps (por sus siglas en inglés *Frame Per Second*) que tiene una cámara.

Redes visuales

Las redes de visión son un conjunto de cámaras cuyo propósito es adquirir imágenes de un mismo proceso para un posterior análisis.

Estereoscópica

Se habla de visión estereoscópica o visión en estéreo cuando se realiza un análisis en dos imágenes a la vez tomadas desde puntos diferentes sobre un mismo objeto o superficie para análisis espacial o crear ilusión de profundidad.

Convergencia

Punto en el cual intersecan los ejes ópticos entre el plano de captura y el plano del objeto.

Acomodación

Enfoque óptico de un punto utilizando cámaras.

Fusión

Proceso conjunto de acomodar un punto y hacer converger los ejes ópticos en él moviendo las cámaras físicamente.

Imagen estereo.

Llamaremos imagen estéreo a la fusión de la imagen del ojo derecho con la imagen del ojo izquierdo o de la imagen derecha e izquierda.

Capítulo 1

1. INTRODUCCIÓN

La percepción del espacio y el mapeo en 3 dimensiones ha sido por décadas uno de los objetivos de la sensorización; la industria de la investigación ha hecho innumerables métodos de métrica a distancia: láser, infrarrojo, ultrasonido, sonares, etc., cantidad de dispositivos para el mismo objetivo, esto es, censar en el espacio y cada uno de ellos ha tenido un limitante.

La naturaleza ha dotado a la mayoría de los seres vivos con un par de ojos, estos son nuestros sensores de visión y con ello podemos obtener imágenes; cada uno de ellos obtiene una imagen de una escena del mundo con un ángulo ligeramente diferente, estas dos imágenes tendrán muchas cosas en común pero cada una de ellas contiene información que la otra no la tiene.

La imagen capturada por cada ojo se envía al cerebro para que éste interprete la información por separado; es mal concebido que las imágenes sean convertidas en una sola; el cerebro interpreta esta información y tenemos la idea de ver una sola imagen pero muchas veces es la captada por un solo ojo la que usamos. Esta información interpretada se conoce como mapa en estéreo¹, así percibimos la profundidad, lejanía y cercanía de objetos, perspectiva y demás características. Este proceso de fusión se denomina **estereopsis**².

¹ *Visión por Computador* Luis Baumela <http://www.dia.fi.upm.es/~lbaumela/Alcala> Departamento de Inteligencia Artificial Universidad Politécnica de Madrid

² <http://www.neoyet.com/vision.htm> VISIÓN EN ESTÉREO Juan José Samper Márquez

Gracias a la visión en estéreo podemos ver los objetos como sólidos en tres dimensiones espaciales con altura y profundidad. Es esta percepción de la profundidad y la perspectiva lo que nos permite visualizar las diferentes distancias y volúmenes del entorno.

Una de las cualidades más importantes de la visión en estéreo es la capacidad de medir y percibir el espacio “vacío”¹. Nuestra vida está estrechamente relacionada y necesitada de la visión en estéreo; operaciones básicas como; recoger, lanzar, alcanzar, aplaudir, manejar, construir necesitan obligatoriamente en muchos de los casos la visión en estéreo.

En la robótica y más específicamente en el estudio de la biónica³ esta capacidad de percepción y análisis sería muy útil especialmente para procesos que tengan relación con las mismas actividades que requieren de la visión en estéreo en nuestro caso, a esto se le llama visión artificial estereoscópica⁴.

Sistemas independientes que tengan una percepción de su entorno en función de la visión tendrían una herramienta importante para sus procesos, y con ella podrían dispensar de muchos sensores que dependen actualmente, estos pueden ser entre otros: láser, infrarrojos, de fin de carrera, etc.

1.1. Descripción de proyecto, alcances

Este proyecto de fin de carrera está enfocado a la creación de un algoritmo y prototipo demostrativo, sin embargo el interés que se puede tener en ella es trascendental ya que la ESPE no ha tenido trabajos científicos de incursión en el análisis de imágenes y la tendencia por el momento en cuestiones de robótica y análisis en manufactura CIM se están encaminando en este tema.

³ <http://es.wikipedia.org/wiki/Biónica> 29 de enero de 2008

⁴ [http://seneca.cucea.udg.mx/curso_linea/Sistemas%20Inteligentes/Modulo%201,%20Seccion%205%20\(1\).pdf](http://seneca.cucea.udg.mx/curso_linea/Sistemas%20Inteligentes/Modulo%201,%20Seccion%205%20(1).pdf) Antonio Reynoso Lobato 2002B CUCEA 29 de enero de 2008

El mapeo tridimensional del espacio, la ubicación específica de objetos, tracing, routing, seguimiento de objetos, son alguna de las funciones que se pueden realizar utilizando el algoritmo que se pretende crear en este proyecto y que se pueden implementar en el campo científico e industrial.

Como producto final de trabajo se espera obtener un algoritmo aplicable a un sistema de visión estereoscópica y demostrar su funcionabilidad con un modelo de red estereoscópica.

El sistema tiene como objetivos principales: diseñar e implementar un algoritmo de ubicación de objetos por medio de una red de visión estereoscópica; también está el de hacerlo eficiente, escalable y de rápido funcionamiento con el objeto de acoplarlo a otros sistemas y mejorar la capacidad de funcionamiento con diferentes redes de visión así como diferentes cámaras.

1.2. Conceptos principales

Dentro de este proyecto de grado se hablará sin detalle de algunos términos que se explican a continuación:

1.2.1. Visión (artificial)

Es la adquisición continua de imágenes por medio de un dispositivo. Esta se la hace a razón de una tasa de adquisición de imágenes con el objetivo de programar un computador para que "entienda" una escena o las características de una imagen.⁵

1.2.2. Tasa de adquisición de imagen

Es la cantidad de cuadros por segundos o fps (por sus siglas en inglés *Frame Per Second*) que tiene una cámara.⁶

⁵ http://es.wikipedia.org/wiki/Visi3n_artificial Visi3n artificial 29 de enero de 2008

⁶ *Visi3n por computador* Arturo de la Escalera Prentice Hall primera edici3n pg: 44

1.2.3. Redes visuales

Las redes de visión son un conjunto de cámaras cuyo propósito es adquirir imágenes de un mismo proceso para un posterior análisis. ⁷

1.2.4. Estereoscópica

Se habla de visión estereoscópica o visión en estéreo cuando se realiza un análisis en dos imágenes a la vez tomadas desde puntos diferentes sobre un mismo objeto o superficie para análisis espacial o crear ilusión de profundidad. ⁸

⁷ *Visión por computador* Arturo de la Escalera Prentice Hall primera edición pg: 54

⁸ <http://en.wikipedia.org/wiki/Stereoscopy> 29 de enero de 2008

Capítulo 2

2. EQUIPOS UTILIZADOS EN VISIÓN ARTIFICIAL

La visión artificial es en definitiva un proceso, que tiene 3 bloques principales, una entrada, un bloque de procesamiento y una salida; siendo un proceso destinado a la visión, es comprensible que los principales dispositivos de entrada sean cámaras, la cuales se conectan con el bloque de procesos utilizando una herramienta necesaria que son los protocolos de comunicación y los estándares de imágenes, la visión artificial también tiene como utilidad especial poder disponer de múltiples entradas, debido a esto la disposición de la forma como las múltiples cámaras se conectarán con el bloque de procesos se llaman “redes de visión”.

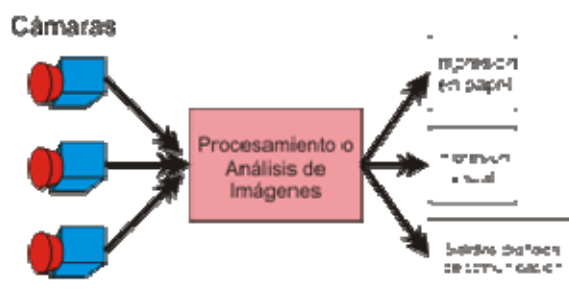


Figura 2.1: Bloque de procesamiento de imágenes

El bloque de procesamiento es en definitiva el computador y el programa que procese las imágenes de las entradas (las cámaras) y la salida esta completamente a disposición de la función a la que se destine el proceso, en nuestro caso la salida sería la posición del objeto a ubicar, esta salida será visual en nuestro caso pero podría ser una trama de datos encapsulada en un protocolo TCP/IP o una salida del puerto serial entre otras.

2.1. Cámaras

Las cámaras son el sensor principal de este proyecto. Al tratarse de un trabajo que analizará imágenes y que necesita una red de visión se entiende que el principal dispositivo para adquirirlas serán las cámaras.

Para ello se necesita entender un poco de lo que una cámara es y para ello dividiremos el análisis en dos partes: la captura de la imagen será la primera y la segunda, la óptica con la que se realiza este proceso.

La captura de la imagen es la parte concerniente al espacio en donde se encuentra el objetivo a capturar; analizo la figura: 2.2:

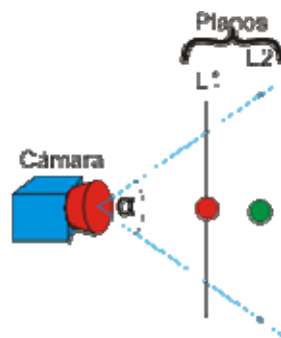


Figura: 2.2 Planos de enfoque

En esta imagen podemos ver que la cámara tiene un ángulo de captura de imagen llamado alfa (en letra griega); esto está representado en las líneas de color celeste, todo lo que se encuentre en esta área estará sujeto a captura, es decir que aparecerá en la imagen.

Las imágenes que captura la cámara tiene una característica de enfoque; debido a las capacidades para adquirir imágenes remitentes a la óptica utilizada para este proceso, generalmente se habla de enfoque de imagen cuando los objetos a interés están bien definidos sobre todo en sus contornos. En la figura 2.2 se puede observar que existen dos círculos: el círculo en rojo se encuentra en el plano L1 y el círculo de color verde está en el plano L2 en una posición más

atrás. Generalmente cuando se enfoca a un plano, por ejemplo L1 no se definen bien los objetos en otros planos, como en el plano L2.⁹

Las características de enfoque podríamos representarlas en la figura: 2.3, en ella podemos ver que existen dos objetos, el de color rojo está enfocado y tiene sus contornos definidos, el de color verde esta desenfocado y no tiene contornos definidos, incluso existen distorsiones en su color y forma.

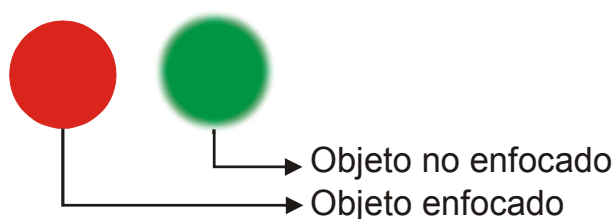


Figura: 2.3 Desenfoco de objetos

Esto ocurre generalmente cuando se usa cámaras con distancias focales grandes; esta distancias es la que existe entre el lente y el punto focal, esto se verá a más detalle en el siguiente capítulo que trata de la óptica de una cámara pero por el momento se puede hablar que el enfoque generalmente está definido a un solo plano de imagen como L1 y L2. Si está enfocado al primero, la imagen del objeto (el círculo rojo) estará definida y en el segundo plano el objeto (circulo verde) está desenfocado.

Igualmente el ángulo de captura no debería variar con el enfoque pero debido a la geometría de los mecanismos para este procedimiento en la mayoría de las cámaras este ángulo se ve distorsionado en pequeña cantidad debido al movimiento que debe tener la lente al momento de enfocar, esto se ve a continuación; en cámaras con distancias focales grandes (o cámaras teleobjetivos) el movimiento del lente es mayor por ello tienen un margen más grande de variación del ángulo de captura que las cámaras que tienen distancias focales pequeñas (cámaras gran angulares) porque para el enfoque este movimiento del lente pequeño, pero para este proyecto dicho parámetro no es de

⁹ <http://www.definicion.org/enfocar> 29 de enero de 2008

gran importancia debido a la pequeña variación que este ángulo representa para estos cálculos (menor a los 2 grados en cámaras sobre los 200mm de distancia focal).

Conviene definir la distancia focal ya que fue un término usado anteriormente y debería quedar aclarado. La distancia focal es como se dijo antes, la distancia que existe entre el lente (en la mayoría de las cámaras es el lente más exterior) y el foco o punto focal.¹⁰

En una cámara simple es como se muestra en la figura: 2.3; en ella se observa la geometría de un lente convexo que generalmente es el que usan las cámaras de un solo lente.

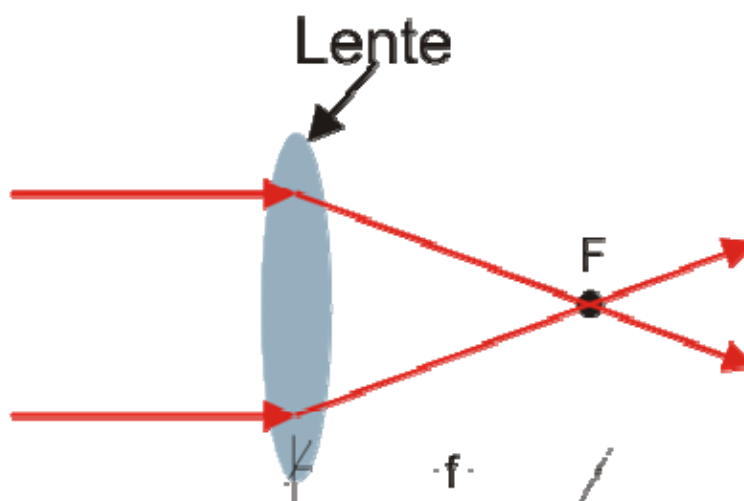


Figura: 2.4 Geometría de cámaras de un solo lente

En la figura 2.4 el foco (F) es un pequeño agujero detrás de la lente con respecto a la imagen a capturar, la distancia focal (f) es la distancia que hay de la lente al foco; la lente converge la imagen por medio de este agujero y esta se proyecta en una superficie atrás del foco con respecto al lente en un plano llamado “plano de impresión”. En las cámaras fotográficas de película se encuentra precisamente la película emulsionante y en las cámaras digitales se encuentra el dispositivo de captura CCD (de esto se hablará posteriormente).

¹⁰ http://es.wikipedia.org/wiki/Distancia_focal 29 de enero de 2008

El enfoque es una pequeña variación de la distancia focal de tal manera que la imagen proyectada sobre la película o el CCD esté definida. En la figura: 2.5 se muestra este proceso.

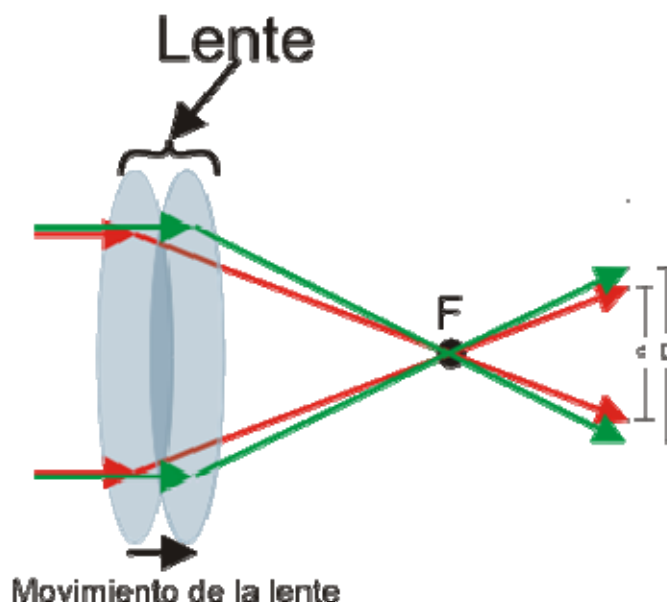


Figura: 2.5 Efecto de enfoque

En esta figura se muestra la geometría del enfoque. Los vectores rojos después de pasar por el lente convergen en el foco e imprimen una imagen de tamaño d ; los vectores verdes hacen lo propio pero al estar el lente más cerca imprimen una imagen de tamaño D . El objeto cualquiera sea debe imprimir un tamaño fijo en el plano de captura, así que el enfoque que debe haber es exacto para ese objeto a esa distancia.

En el plano de captura se encuentra el dispositivo que como antes se mencionó es la película emulsionada o el dispositivo digital de captura.

Los dispositivos digitales de captura son de dos tipos: los CCD (*charge-coupled device*: "dispositivo de cargas [eléctricas] interconectadas") y los CMOS (*Complementary Metal Oxide Semiconductor*: "Semiconductor Complementario de Óxido Metálico"). Los CCD tienen claras ventajas sobre los dispositivos CMOS en cuanto a calidad de imagen, inmunidad a ruidos, definición y calidad, pero en el

transcurso del tiempo los dispositivos CMOS están ganando terreno, debido principalmente a su capacidad de trabajo matricial (se puede acceder a sectores de la captura independientemente), inmunidad a ciertos efectos adversos (destellos de puntos luminosos en los píxeles vecinos “blooming”), están hechos de la misma manera que los procesadores así que puede integrarse al procesamiento en el mismo chip y utilizan apenas 5V frente a los 10-30V de los CCD.¹¹

Los dispositivos de captura de imagen digital han alcanzado y rebasado en algunos casos a las películas especiales, así de esta manera se tienen cámaras digitales con características como:

2.1.1. Visión: Las cámaras de visión son las cámaras que generalmente usamos, como ejemplos principales tenemos las cámaras de video, fotográficas y webcams, estas tienen un CCD o CMOS que captura las imágenes en un espectro muy cercano al que podemos ver como humanos. Estas cámaras capturan imágenes que se desean observar. (fotografías o videos), sin embargo el análisis en este tipo de imágenes no siempre es óptimo.

En razón que las cámaras son más usadas, la fabricación de estos dispositivos han hecho que su precio baje significativamente lo que ha conllevado que la mayoría de análisis se los haga en estas.

2.1.2. Infrarrojas: Las cámaras infrarrojas son cámaras que capturan las imágenes en ese espectro; estas cámaras tienen una ventaja marcada en el momento de hablar de iluminación. Cuando iluminamos un objeto en espacio abierto, dígame un jardín, balcón o cualquier exterior, estamos a merced del sol o de la iluminación del lugar, incluso en interiores se necesita iluminación del lugar para el deambular de la gente.

¹¹ [http://es.wikipedia.org/wiki/CCD_\(sensor\)](http://es.wikipedia.org/wiki/CCD_(sensor)) 29 de enero de 2008

Cuando esto ocurre es un problema si queremos iluminación frontal de los objetos, entonces podemos usar iluminación frontal infrarroja y hacerla intensa, usar una cámara infrarroja y capturar la imagen; si la fuente lumínica es intensa la imagen es nítida, la luz exterior o ambiental no afectará la captura de la imagen, incluso si se puede controlar las demás iluminaciones, la captura podría ser algo parecido a la perfección.

La captura de imágenes a color se hace por medio de 3 sensores en escalas de grises; un prisma divide la imagen en tres colores y cada imagen es recibida por cada uno de los sensores, así se puede hacer una tricromía de imágenes para imprimir el color como las impresoras offset. En la figura: 2.6 se puede observar este proceso:

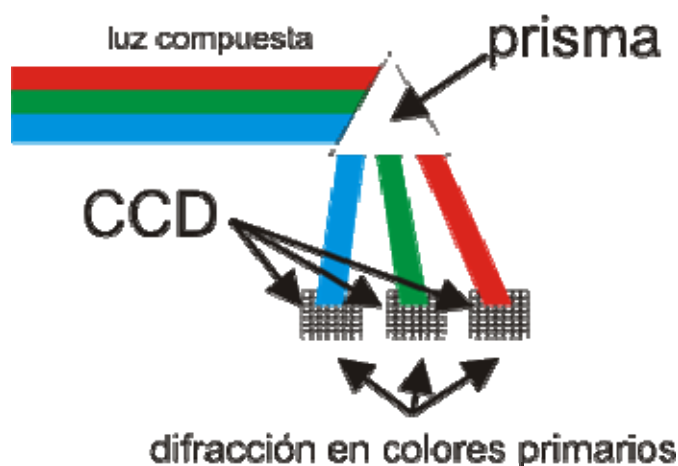


Figura: 2.6 Difracción de la luz en cámaras

En cámaras como las infrarrojas la distorsión de color se hace evidente porque no existe la división de colores y la mayoría de superficies no reflejan del mismo modo los colores normales que la luz infrarroja; cuando se quiere observar esta imagen generalmente se hace una transformación de la escala infrarroja a la escala de los grises, allí es donde la imagen se puede ver distorsionada, si bien es una desventaja al momento de observar la imagen se debe tener en cuenta la gran importancia de su capacidad de adquisición de imágenes

2.1.3. **Más cámaras:** Las cámaras también tienen una característica sobre sus formatos de captura, generalmente estamos acostumbrados a

recibir imágenes en proporciones que son 4/3, es decir que es cuatro veces más ancha que 3 veces en alto. Existen los nuevos formatos como el 16/9 que tiene una proporción de 16 veces en ancho a 9 en alto, estas por supuesto son más anchas.

Pero quizá las cámaras con formato diferente más usadas son las lineales; son cámaras que capturan la imagen en una sola hilera, pero no son categorizadas como cámaras porque estamos acostumbrados a percibir las imágenes en forma matricial. El ejemplo más claro de éstas son los scanner de documentos; estos son cámaras que capturan la imagen en una sola línea y así digitalizan los documentos impresos.

También se observan estas cámaras en los supermercados, las cajas registradoras tienen este tipo de cámaras para capturar los códigos de barras de los productos; éstas son cámaras lineales que trabajan en el espectro infrarrojo, aprovechando una iluminación extra infrarroja intensa proporcionada por un láser.¹²

2.2. Protocolos

Si bien los protocolos no son equipos, son herramientas muy útiles en el campo de la adquisición de imágenes, estos hablan directamente de la velocidad, calidad, facilidad, compresión y formatos en los que las cámaras se comunican con los equipos de análisis o procesamiento de imágenes.

El procesamiento de imágenes es en definitiva cuando el producto de un trabajo sobre una imagen es otra imagen; el análisis en cambio es cuando esto no ocurre, es decir que se realiza un trabajo sobre una imagen y el resultado es algo que no es una imagen: por ejemplo cuando editamos fotos para recortarlas, aclararlas, girarlas o en definitiva mejorarlas. Entonces estamos procesando imágenes cuando leemos el código de barras de un producto o digitalizamos las

¹² *Visión por computador* Arturo de la Escalera Prentice Hall primera edición pg: 62

letras de una imagen o ubicamos objetos, allí estamos haciendo análisis en imágenes.

En cuanto a la velocidad, calidad y compresión podemos decir que esta está en relación directa al ancho de banda que se disponga; mientras más “ancho de banda” se tendrá más velocidad y calidad y, menos compresión será necesaria.

El protocolo está formado de dos partes, la primera es el bus de transmisión de datos y la segunda, es el proceso de compresión de las imágenes.

Entre los buses más usados tenemos el FireWire, USB y Camera Link¹³ debido a su facilidad y popularidad, así como por las características y beneficios de cada uno de ellos es un campo muy largo pero se puede enfatizar algunos puntos que los hacen los mejores en cada uno de sus casos.

El bus FireWire es muy usado en la actualidad por el bajo costo que han adquirido sus repetidoras y hub's. Hace algunos años estos equipos eran un poco costosos en comparación de la popularidad de los comparables equipos en USB; FireWire es un bus mucho más complejo, es multimaster lo que significa que múltiples equipos pueden pedir información al periférico, es decir que si muchas computadoras están conectadas a la misma cámara y mientras el bus de datos lo soporta, todas ellas pueden acceder a los datos de la cámara.

El bus USB en cambio tiene la característica de ser monomaster, lo que quiere decir que el periférico debe estar conectado a un solo computador y solo este puede acceder a los datos; las ventajas propias del bus son la gran cantidad de equipos que podemos conseguir, esto abarata los costos y las implementaciones de equipos.

El bus CameraLink es muy usado en NVR (*network video recorder*: “grabadores de video en red”) y DVR (*digital video recorder*) que generalmente se

¹³ *Image processing with LabView and IMAQ Vision* Thomas Kilinger Prentice Hall PTR pg:2 cap3

usan en sistemas CCTV (Circuito Cerrado de Televisión) para seguridad, tiene la peculiaridad de tener un bus independiente para cada cámara pero generalmente las tarjetas de adquisición de video soportan muchas en proporciones binarias como 8/16/32 cámaras a la vez y sus definiciones son muy superiores en la mayoría de los casos.

En el cuadro 1 podemos observar las ventajas de cada protocolo para las capacidades en transmisión de video.

En cuanto a la compresión existen muchos algoritmos para tipos de compresión pero principalmente se dividen en compresiones con pérdida y sin pérdida; esto quiere decir que cuando se hace la compresión se pierde en parte un poco de información.

Entre los algoritmos de compresión sin pérdida están: codificación de longitud de corrida, codificación Fuman, codificación Lempel-Ziv y codificación aritmética; estas codificaciones son complicadas y meramente matemáticas. Si bien es mejor no perder absolutamente nada de la información, hay partes que ni siquiera podemos utilizarlas como frecuencias de luz muy altas o muy bajas que el ojo humano no puede percibir o incluso variaciones de tonalidades que no podemos percibir; enumeramos estas compresiones debido a que el lector puede sentirse atraído a investigar mas a fondo estos algoritmos.

	IEEE 1394 FireWire	USB 2.0	Camera Link
Tasa máxima de transferencia	800Mbit/s	480Mbit/s	~7.14Gbit/s
Arquitectura	Peer a peer	Orientado a host	Peer a peer
Dispositivos máximos	63	127(incluido hub's)	-
Máxima longitud del cable	4.50m	5m	~10m
Distancia máxima posible	72m	30m	~10m
Cables para 8bits por píxel	4	2	10
Comunicación bidireccional	Si (asíncrona)	Si (asíncrona)	Canal adicional
Entrega de poder	8-40V, 1.5A	5V 500mA	-

Cuadro: 2.1 Protocolos de comunicación¹⁴

Los algoritmos de compresión con pérdida se puede clasificar en cuanto a su uso para: conferencias, audio, imágenes y video; entre las compresiones para video tenemos: ITU-T 4 y T6, JBIG (ITU-T T82) y JPEG (ITU-T T.81), la mayoría usan transformada del coseno pero últimamente están usándose las transformadas wavelet que está dando buenos resultados para las compresiones JPEG 2000

En las compresiones para video se tiene ITU-T H28 y MPEG, ésta última es una compresión basada principalmente en la compresión de imágenes JPEG; de la misma manera MPEG2000 usa transformada wavelet para mejorar la compresión.

Los algoritmos para compresión con pérdidas están hechos de tal manera que se pierdan características que no se puede ver a simple vista, como tramas complejas y minúsculas, ruido, colores y degradaciones imperceptibles para el ojo humano, blooming, etc. El objetivo es perder información que no podamos darnos cuenta que ha desaparecido, incluso en ciertas circunstancias se podría decir que hay una mejoría.

¹⁴ *Image processing with LabView and IMAQ Vision* Thomas Kilinger Prentice Hall PTR pg:27 cap3

Entrar en materia de estas compresiones o de los buses no compite para este proyecto pero si es bueno saber las características de cada uno para saber sus posibilidades y capacidades en el momento de escoger y trabajar con ellas.¹⁵

Como ya se ha visto, las imágenes son matrices de puntos, la más simple de todas sería una imagen en blanco y negro, estas tienen un bit por cada píxel; las cámaras de escala de grises tienen de 8 a 32 bits generalmente para la cantidad de patrones de gris que podrían presentar, mientras más bits más grises se pueden representar. Por otro lado, las imágenes a color están formadas por una tricromía de imágenes o incluso cuatricromía en ciertos casos.¹⁶

Las tricromías de imágenes digitales están formadas por 3 patrones: los colores luz RGB (rojo, verde y azul) forman colores combinándose y en conjunto forman el color blanco en la saturación de colores; los colores tinta o CMY (cian, magenta y amarillo) también forman los colores combinándose pero la saturación de estos tres en conjunto forman el color negro; otra tricromía es el HSL (matiz, saturación y luminancia) que forma colores típicamente para los equipos televisivos.

La cuatricromía más usada es la CMYK la cual incluye el color negro a los colores tinta debido a que estas combinadas no producen un color negro saturado en las imágenes impresas; los diseñadores usan esta cuatricromía debido a la imperfección de las impresoras en su proceso de tinturación de la imagen, debido a esto generalmente también se calibran los monitores en función de impresoras para que los usuarios vean en verdad que colores se imprimirán.

Cuando se diseña algo que va a ser proyectado se usan la tricromía RGB porque los proyectores usan esos mismos colores para proyectar la imagen; si la imagen va a ser impresa se usa CMYK y si se va a presentar en un monitor CRT (tubo de rayos catódicos en siglas en inglés) se usa HSL porque de esa manera es como presentan las imágenes en las pantallas.

¹⁵ *Image processing with LabView and IMAQ Vision* Thomas Kilinger Prentice Hall PTR pg:28 cap3

¹⁶ *Visión por computador* Arturo de la Escalera Prentice Hall primera edición pg: 60

En la figura: 2.7 se puede ver algunos ejemplos de lo expresado en este numeral.

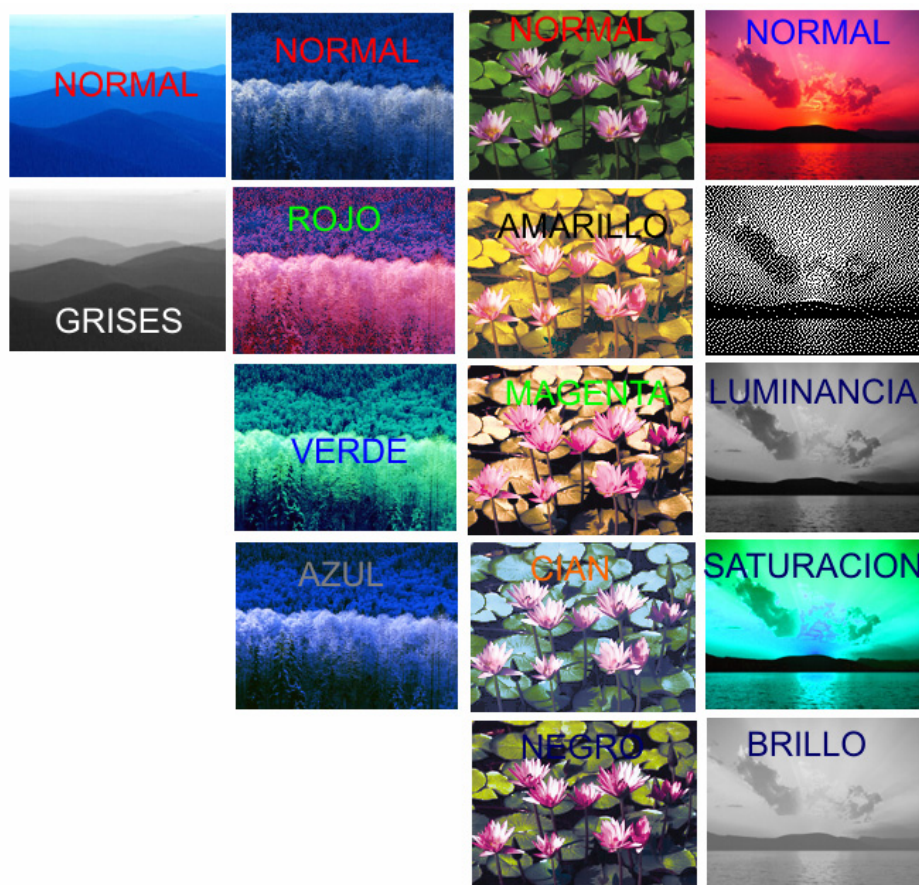


Figura: 2.7 Imágenes por canal

2.3. Redes de Visión

Las redes de visión son un conjunto de cámaras necesarias para que cumplan una función determinada, esta puede ser por ejemplo: seguridad en una empresa en donde existen muchas cámaras y de gran definición pero no se necesitan muchos cuadros seguidos o un sistema de control de calidad en donde se necesita generalmente una sola cámara pero muchos cuadros por segundo para mejorar la velocidad de la verificación; debido a estos requerimientos se escogen los protocolos, las topologías y las cámaras.¹⁷

¹⁷ http://es.wikipedia.org/wiki/DVC_codificación_distribuida_de_vídeo 29 de enero de 2008

El caso más típico de una red de visión es una cámara de escritorio USB, esta se compone de una cámara de tipo CMOS, generalmente son simples, su compresión de imágenes está a cargo del algoritmo de compresión MPEG y el bus que usa por supuesto es el USB.

Un caso más complejo de redes de visión corresponde a las cámaras de seguridad, estas redes están formadas por un conjunto de cámaras cuyas características especiales son tener un rango más amplio de visión en cuanto a luminosidad; un ejemplo claro de esto es una cámara que vigilará en un hall de establecimiento: en el día esta muy iluminado debido al sol, en la noche la luminosidad es baja o quizá hasta inexistentes; debido a esto se usan cámaras híbridas que usan tanto sensores para el espectro luminoso visible y para el infrarrojo. Estas cámaras poseen unos led's de alta luminosidad en el rango infrarrojo para que cuando se detecte poca luminosidad brinden luz para el lugar y se pueda capturar imágenes infrarrojas.

En una red de visión para seguridad también cambia la cantidad de cámaras, los lugares más complejos usan cientos de cámaras, por ello el bus de transmisión de datos es generalmente Camera Link; este es mucho más robusto para transmisión de imágenes de múltiples cámaras como también soporta un mayor número de ellas En la mayoría de los casos tampoco se comprimen las imágenes con pérdidas, generalmente se usan compresiones aritméticas.

Para el caso del observador, controlar esta cantidad de cámaras es un absurdo y contratar un número mayor de observadores es un desperdicio, por ello se desarrollan algoritmos muy complejos para análisis de imágenes e incluso existen sistemas que pueden detectar por ejemplo en una imagen cuando una persona se encuentra en peligro.

Las redes de visión incluyen el escoger el tipo de cámara, la topología del bus y el tipo de compresión que deba utilizarse; existen redes análogas a las redes de computadoras, es decir, token-ring, estrella, mixtas, nebulosas, etc. Un ejemplo claro y su semejanzas se puede ver en la figura: 2.8

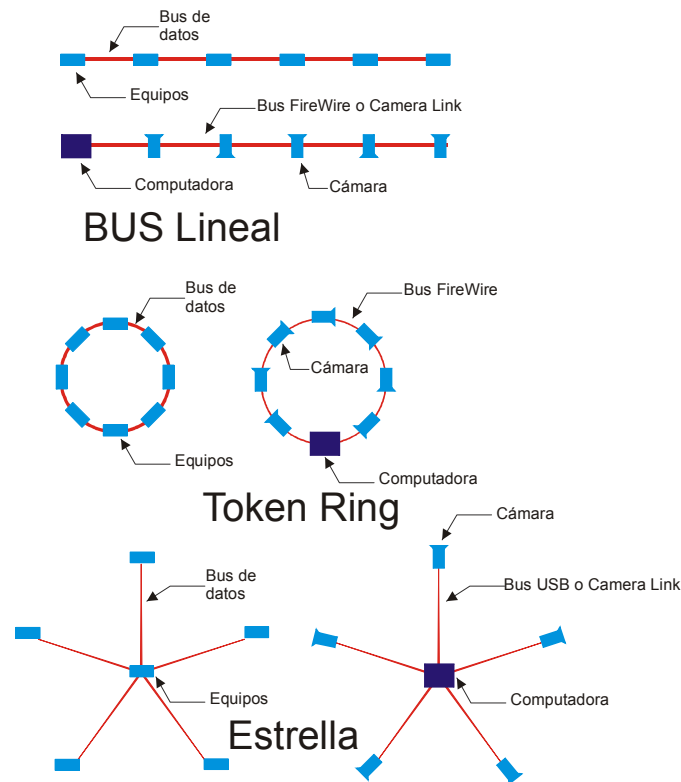


Figura: 2.8 Topologías de redes

En este gráfico se muestra las topologías principales que existen, el bus lineal es básico. El bus FireWire está diseñado para que todos los equipos que se conectan sirvan de repetidores para los que se acoplen a ellos por lo que lo convierte en el bus preferido para esta topología.

La topología de Token Ring ayuda a que los equipos tengan dos vías por donde comunicarse, es más robusta pero necesita un extra de cable para conectarse y un protocolo más complejo que identifique el mejor camino para la comunicación.

La conexión en estrella está dominada por USB; esta es básica y lo bueno es que si un equipo falla los demás no lo hacen ya que cada uno tiene su propia

vía de comunicación. Si bien FireWire también puede implementar esta topología, USB es la que predomina, además de la topología Camera Link.¹⁸

¹⁸ http://es.wikipedia.org/wiki/Bus_de_datos 29 de enero de 2008

Capítulo 3

3. ESTEREOSCOPIA

3.1. Definición

La visión estereoscópica o estereoscopía es la capacidad de analizar y procesar dos imágenes diferentes con el objeto de tener una disparidad, ésta es la diferencia que se obtienen al tener dos imágenes de una misma escena tomada de dos ángulos diferentes al mismo tiempo.¹⁹

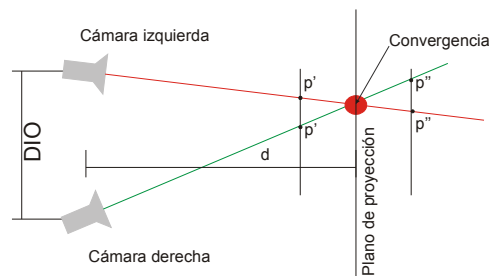


Figura: 3.1 Planos de observación con giro de cámara

En la figura: 3.1 se puede observar los principales elementos de la estereoscopía artificial y de entre ellos las cámaras son lo primero; generalmente al hablar de estereoscopía se habla de cámaras cuya disposición es horizontal, por ello existe una cámara izquierda y otra derecha pero no es un caso omiso que otra disposición sea vertical y se tenga una superior y otra inferior; el hecho de tener una disposición horizontal se debe a la mayoría de casos en donde los objetos tienden a tener un movimiento más errático horizontalmente debido a que

¹⁹ <http://en.wikipedia.org/wiki/Stereoscopy> 29 de enero de 2008

verticalmente están sujetos a los efectos de la gravedad que solo los mueve hacia abajo y dado que pocos de ellos tienen la capacidad de volar se supone que sus movimientos serán horizontales en la mayoría de casos.²⁰

En la figura: 3.1 se ve también una distancia entre las dos cámaras llamada distancia interocular (DIO); este quizá es el parámetro más importante dado que mientras mayor sea la DIO se tendrá más disparidad y mayor captación de profundidad en objetos lejanos y a menor DIO se tiene menor disparidad pero mayor captación de profundidad en objetos cercanos. Este particular será tratado mas adelante.²¹

También se tiene una distancia d , llamada *distancia al centro de proyección* que va desde el centro geométrico entre las dos cámaras hasta el centro del plano de proyección, este es paralelo al plano de las cámaras y es en donde se encuentra el objeto, aquí el punto de convergencia.²²

El paralaje es la distancia que hay entre dos proyecciones de un punto; en el plano de proyección, en la figura: 3.1 se ve que en el primero existen los puntos p' , estos son la proyección del punto de convergencia en un plano negativo, por ello se llama *paralaje negativo*, igualmente existen los puntos p'' que son el producto de un paralaje positivo, si el paralaje es cero se tiene al objeto y el plano de proyección en el mismo lugar.²³

En la estereoscopia existen dos formas de ubicar a un objeto: el primero es coincidir el plano de proyección con el objeto, es decir que el paralaje tenga un ángulo de cero grados, de esta manera si se mide los ángulo de las cámaras y su distancia intraocular se puede hacer triangulación y así ubicar al objeto; la segunda forma es no girar las cámaras de ninguna manera, el paralaje entonces tiene la misma magnitud de la DIO, es decir que las proyecciones de las cámaras es al infinito y no se juntan, de esta manera se puede ubicar al objeto, a saber, el ángulo que tiene el objeto en la imagen con respecto a la normal; esta es la

²⁰ <http://en.wikipedia.org/wiki/Stereoscopy> 29 de enero de 2008

²¹ <http://www.users.red3i.es/~stereoweb/vision.htm> 29 de enero de 2008

²² <http://www.neoyet.com/vision.htm> VISIÓN EN ESTÉREO Juan José Samper Márquez

²³ <http://es.wikipedia.org/wiki/Paralaje> 29 de enero de 2008

manera de cómo se ubicarán los objetos en este proyecto. En la figura: 3.2 se puede visualizar lo explicado.

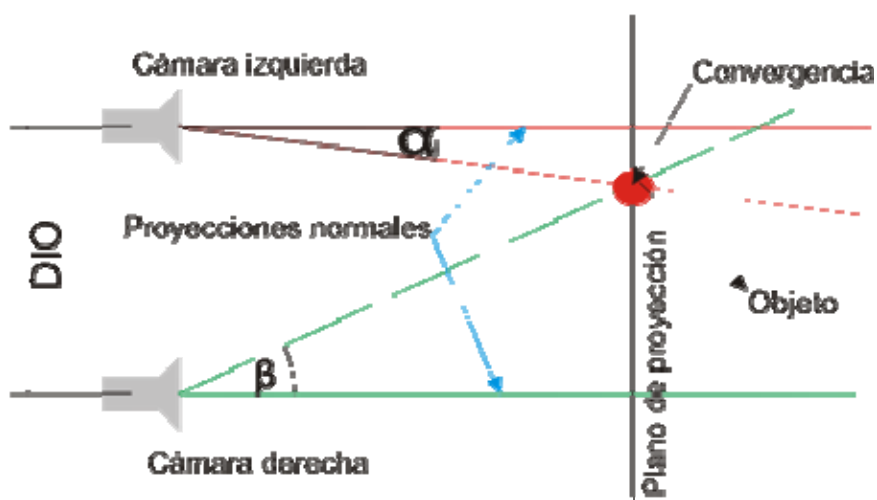


Figura: 3.2 Planos de observación con giro de cámara

Aquí el ángulo alfa y beta puede ser encontrado en la imagen de cada cámara, de esta manera la triangulación puede ser hecha sin girar las cámaras de ninguna forma pero las condiciones están relacionadas directamente con otros parámetros como: el *ángulo sólido de captura*.

El ángulo sólido de captura no es más que el ángulo total horizontal que tiene una cámara el momento de capturar la imagen; luego el ángulo vertical puede ser calculado tomando las proporciones de la imagen. Así, si una cámara captura en proporción $4/3$ entonces quiere decir que el ángulo vertical será tres veces la cuarta parte del ángulo horizontal; si la proporción es $16/9$ entonces sería nueve veces la dieciseisava parte del ángulo horizontal.²⁴

²⁴ http://es.wikipedia.org/wiki/Ángulo_de_visión 29 de enero de 2008

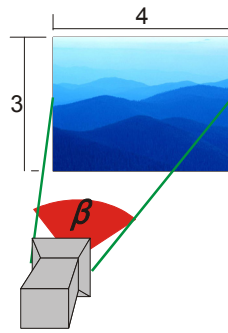


Figura: 3.3 Ángulo sólido de visión

Ahora se puede calcular también el ángulo con respecto a la normal, es decir, si examina al objeto dentro de la imagen y se conoce el ángulo sólido de captura, entonces es fácil dividir la imagen en ángulos para cada píxel y así reconocer en ángulo tanto de elevación como el de giro.

Como se ve en la figura: 3.4, el objeto a ubicar se encuentra entre píxel, así que el objeto es ubicado entre ellos y el centro es calculado con el método de masas mas no el geométrico.

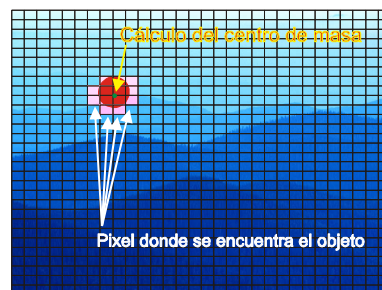


Figura: 3.4 Redondeo de píxeles en la captura de imágenes

La diferencia de calcular el centro de un objeto usando el centro de masas radica en la posible forma de este; esto se ve claramente en la figura: 3.5 en donde un objeto tiene más partes cubriendo píxel en un lado de la imagen que de otro y así su centro geométrico difiere de su centro de masas.

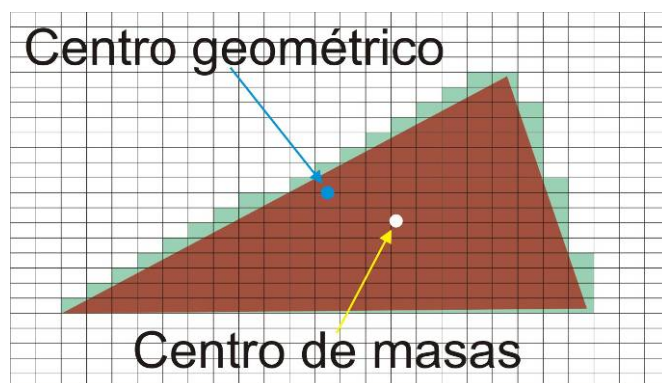


Figura: 3.5 Centro de masas en objetos digitalizados

En esta figura se muestra claramente como el centro geométrico difiere del centro real por estar ubicado entre los extremos de la figura mientras que el centro de masas pondera la cantidad de píxel que hay a uno u otro lado para calcular su centro; este podría ser el centro real de la figura si su densidad es constante y si su forma es de ancho constante al plano de proyección; si no es así, éste podría diferir mucho del real.

3.2. Razones para el uso de estereoscopia

Si bien la estereoscopia parecería la forma ideal para ubicar objetos, esta tiene sus desventajas; una de sus principales es la necesidad de iluminación que no tienen otras como el láser o el sonar, también es susceptible a ruido como niebla y humo. Las cámaras CCD o las CMOS tienen también un problema con respecto al ruido térmico y todavía no se ha podido eliminar efectos de distorsión como el blooming, destello, alto contraste, barril, ojo de pez entre otros que son producidos principalmente por la estructura; el sensor o por la forma de los lentes. Esto distorsiona de tal manera la captura de la imagen transformándose en un error significativo de ubicación. Esto no es un problema del algoritmo. La ubicación estereoscópica es muy exacta si dependiera solo de los cálculos, el problema son las cámaras.²⁵

²⁵ <http://www.neoyet.com/vision.htm> VISIÓN EN ESTÉREO Juan José Samper Márquez

3.3. Posibles soluciones con algoritmos

Las mejores soluciones para las distorsiones de forma como el barril, ojo de pez, miopía, es la ubicación con rotación, ya que el centro normal de la captura de la imagen sigue siendo perfecta.

Es decir, como se mostraba en la figura: 2.2, se hace coincidir los centros de las capturas de las cámaras con el centro de masas del objeto y con un ángulo de rotación se ubica al objeto; es de esta manera como se dispone de los objetos a diario.

Esta solución parecería ideal de no ser por el problema del tiempo de reacción de los motores y actuadores; estos generalmente son neumáticos ya que son más rápidos pero su costo debido a la exactitud que se necesita es muy elevado. Los motores eléctricos de pasos son un poco lentos pero exactos, los servomotores son fuertes pero lentos, en todo caso, para cualquiera de ellos hay pasos intermedios que necesitan muchos procesos. Así un objeto a ubicar primero es encontrado en la imagen y se calcula su posición con respecto al centro, se gira la cámara y se vuelve a hacer el proceso de ubicación, nuevamente se ve su error con respecto al centro y se vuelve a girar hasta que el error se aproxime o sea cero; este proceso se ve en la figura: 3.6.

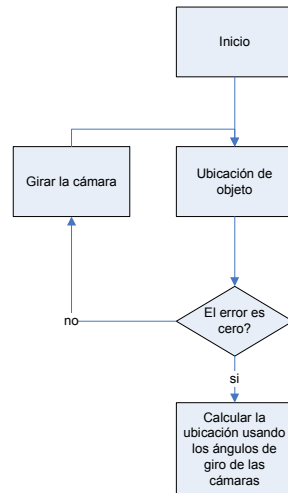


Figura: 3.5 Algoritmo de posicionamiento de cámaras

El lazo del diagrama de la figura: 3.5 que va desde la pregunta “¿El error es cero?”, pasando por la respuesta “no” es lo que hace al proceso lento, cada vez que pasa por el paso de ubicación de objeto se está haciendo un cálculo fuerte en cada imagen; este paso es el que ubica en gran parte al objeto sin girar las cámaras, esto se ve en la figura 3.6.

Contra las distorsiones de los sensores solo se puede esperar que mejoren con el tiempo y la tecnología. El blooming ya se esta controlando con sensores que tienen mejores definiciones y que pueden identificar donde se encuentra un destello dentro de la imagen y así evitar la saturación de píxel aledaños y eliminar el efecto.

De la misma manera se puede esperar que los efectos de captura como los de lente desaparezcan o se controlen; así, el algoritmo estereoscópico que se procederá a utilizar sin rotación también sería exacto.

El siguiente algoritmo es el que no gira las cámaras, lo cual puede observarse claramente en la figura: 3.6.

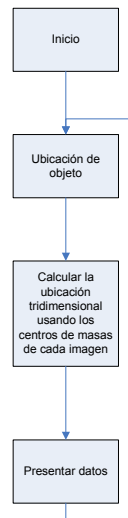


Figura: 3.6 Algoritmo de ubicación sin posicionamiento de cámaras

En esta figura 3.6 se ve como el lazo es recto, no hay una pregunta, simplemente se usa los datos que devuelve cada bloque. Si el bloque de ubicación de objetos devuelve un error simplemente el siguiente continúa con el error hasta que el bloque de presentación de datos devuelva un error. Otra ventaja de este algoritmo es que puede seguir ubicando al objeto mientras se mueve ya que el lazo continua en el primer bloque después del inicio mientras que el anterior algoritmo, una vez finalizado hay que volverlo a inicializar, si lo que se quiere es seguirle al objeto en su movimiento.

Los dos algoritmos no tienen una pregunta para salida o fin del proceso pero para un programa desarrollado se debería incluir no solo estas condicionantes de fin para el proceso sino también condiciones de error que interrumpen el ciclo.

Capítulo 4

4. DISEÑO DEL ALGORITMO

En este capítulo se procederá a diseñar el algoritmo propiamente dicho para luego analizar una forma de hacerlo en un lenguaje de programación.

Como bases principales del algoritmo se tiene que su red de visión deberá cumplir una topología cualquiera así como también cualquier cámara, entre las características que por el momento se pueden anotar se citan:

- La necesidad del ángulo sólido de visión de las cámaras
- La adquisición de cada cuadro debe ser dinámico e ininterrumpido
- La topología de la red de visión debe asegurar el último punto
- Las cámaras deben adquirir imágenes cuya disparidad no sea de fondo

El último punto se debe a la función del algoritmo, como ya se explicó antes, la disparidad de fondo en el caso humano está regulado por el cerebro; incluir esta función en el algoritmo es un paso de mucho proceso, es mucho más fácil regular las cámaras para que su adquisición de datos sea de la misma forma.

4.1. Funciones principales

Entre las funciones principales del algoritmo están:

- Permitir al usuario determinar el objeto a identificar en las imágenes.
- Ser de rápido procesamiento.

- Ser de lazo cerrado y que se rompa el mismo en caso de querer una nueva búsqueda.
- Ajustarse a diferentes topologías y cámaras.
- Ser muy preciso en los rangos que permita la disparidad.

La velocidad de procesamiento se la comparará con el flujo de datos, es decir que si el proceso de ubicación de objetos se demora más del tiempo en que un nuevo cuadro es capturado por la cámara entonces el proceso es lento, dependiendo de la cantidad de tiempo que deje nuestro proceso entre estos lapsos se podrá incorporar más funciones al proceso de análisis de imágenes para optimizar el equipo de análisis ya que al ser destinado a la ubicación de objetos y demorándose menos del tiempo requerido por la cámara para capturar otra imagen, este equipo estaría subutilizado; debido a esto este tiempo será analizado en diferentes configuraciones de hardware.

La precisión está en función a la cantidad de veces que el proceso devolverá un dato y cada vez que lo repita devuelva el mismo dato, no es exactitud lo que se busca aunque luego se demostrará que tan exacto es. La razón principal es que al ser un sensor virtual la exactitud se puede buscar con calibración mas la precisión es un dato mucho más necesario.

Con esto se puede comenzar a enfatizar en los puntos ordenados en los que consistirá el algoritmo pero para esto se resume los casos ya analizados para ciertos puntos de tal manera que se comienza a desechar y adoptar algunos casos para nuestro algoritmo:

La topología deberá ser tipo estrella, debido a la velocidad que tiene esta al no tener un token (paquete) de información que suceda al siguiente en los nodos como en el caso del bus o token-ring; esto requiere que el computador difiera cada paquete y lo ordene, si bien no es un proceso extenuante, sí requiere que exista un espacio de memoria que mantenga el dato que viene en serie por el bus ya que nuestro propósito es la eficiencia y velocidad, una topología estrella requiere más estructura física pero su eficiencia a nivel de procesamiento es evidente.

Las cámaras deben adquirir las imágenes con calidad en la definición, esto va directamente relacionado con el error al momento de calcular la ubicación del objeto por obvias razones.

Asumiendo que la adquisición de las imágenes cumplan con las características de disparidad antes anotadas, un manejo de errores no debería ser necesario a excepción de los errores que pueden encontrarse naturalmente en el cálculo previsto. Estos errores son por ejemplo que el objeto, si se encuentre en la imagen pero con colores diferentes, este error no se contempla en el algoritmo ya que la disparidad solo será de forma.

La interfaz con el usuario debe permitirle identificar al objeto manualmente y de forma intuitiva; debido a que el proceso de ubicación de objetos está en pausa, éste puede ser tan complejo como sea necesario con el objetivo de facilitar al operador mayores herramientas y facilidades.

No se intentará entender al objeto, es decir que no se sabrá de su forma, de su posición con respecto a su normal, de su color o demás características que lo califiquen, solo se intenta identificar su posición con respecto a un punto fijo el cual para nuestro caso será el observador derecho (cámara derecha).

Ya que la ubicación del objeto es espacial y que se necesita un punto de referencia se toma la cámara derecha, la cual en el punto mismo del foco tendrá las coordenadas 0,0,0.

Ya que los cálculos que se realizarán serán hechos por medio de triangulación, lo correcto será utilizar las coordenadas polares.

4.2. Estructura Switch On/Off Y Casos De Excepción:

Lo principal es el algoritmo sin los casos de excepción, es decir que el objeto se encuentra en una posición que permite a la triangulación hacer los cálculos de forma precisa y sin cálculos extras, así se tiene:

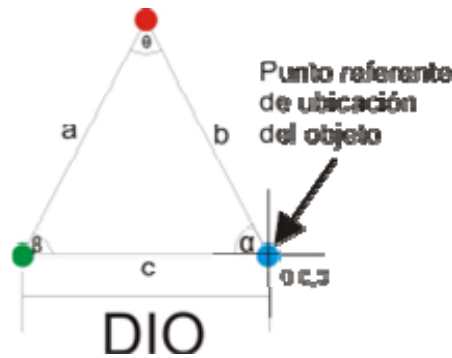


Figura: 4.1 Triangulación simple sin ángulos obtusos

Los ángulos del triángulo internos de la figura 4.1 son α , β , θ y sus lados opuestos son a , b y c respectivamente.

En esta figura existen 3 ángulos y todos ellos son agudos, es decir que no tienen más de 90 grados, en este caso es fácil ubicar las distancias de cada uno de los lados, si lo que se conoce la distancia del lado inferior que antes se le llamó la DIO y los ángulos alfa y beta también son un dato conocido entonces las distancias que faltan se calculan con la ley de senos:

$$\frac{a}{\text{sen}(\alpha)} = \frac{b}{\text{sen}(\beta)} = \frac{c}{\text{sen}(\theta)} \quad (4.1)$$

Si lo que busco es la distancia b debido a que las referencias las buscaré con respecto a la cámara derecha entonces se encuentra así:

$$b = \frac{c \times \text{sen}(\beta)}{\text{sen}(\theta)} \quad (4.2)$$

$$c = DIO$$

$$b = r$$

$$r = \frac{DIO \times \text{sen}(\beta)}{\text{sen}(\theta)}$$

La distancia r es debido a la reducción de la palabra radio y a que las distancias se calcularán con coordenadas polares.

El ángulo θ se calcula con la relación de la suma de los ángulos internos de un triángulo:

$$\alpha + \beta + \theta = 180$$

$$\theta = 180 - \alpha - \beta$$

(4.3)

El problema principal radica cuando se tienen ángulos mayores a 90° o sea obtusos, para esto se necesitan hacer los cálculos con las leyes trigonométricas y dividir al triángulo de la siguiente forma:

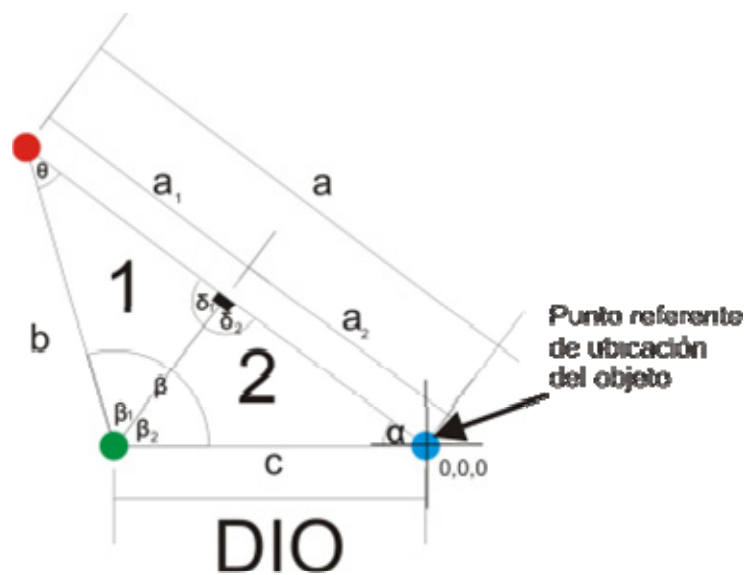


Figura: 4.2 Triangulación con ángulos obtusos

Aquí el triángulo ha sido dividido en dos triángulos rectángulos, con esto el ángulo beta se divide en dos ángulos y se tienen dos ángulos rectos: delta 1 y delta 2, también se divide el lado “a” en dos partes “a1” y “a2”, entonces usando el teorema de Pitágoras tenemos:

$$\begin{aligned}
 \text{sen}(\alpha) &= \frac{d}{c} \\
 d &= \text{sen}(\alpha) \times c \\
 \text{TRIANGULO}_1 \\
 \beta_1 + \beta_2 &= \beta \\
 \beta_1 &= \beta - \beta_2 \\
 \text{tag}(\beta_1) &= \frac{a_1}{d} \\
 a_1 &= \text{tag}(\beta_1) \times d \\
 a_1 &= \text{tag}(\beta_1) \times (\text{sen}(\alpha) \times c)
 \end{aligned}
 \tag{4.4}$$

De esta manera se puede calcular la distancia total obteniéndose los trayectos de las dos partes que conforman los triángulos 1 y 2.

$$\begin{aligned}
 \text{TRIANGULO}_2: \\
 \text{COS}(\alpha) &= \frac{a_2}{c} \\
 a_2 &= \text{cos}(\alpha) \times c \\
 c &= \text{DIO} \\
 a_2 &= \text{cos}(\alpha) \times \text{DIO} \\
 a &= a_1 + a_2 \\
 a &= \text{tag}(\beta_1) \times (\text{sen}(\alpha) \times \text{DIO}) + \text{cos}(\alpha) \times \text{DIO}
 \end{aligned}
 \tag{4.5}$$

$$\begin{aligned}
 180 &= \beta_2 + \alpha + \delta_2 \\
 \delta_2 &= 90^\circ \\
 \beta_2 &= 90 - \alpha
 \end{aligned}
 \tag{4.6}$$

Las distancias están calculadas por medio de trigonometría así que el error de cálculo está solo limitado a la cantidad de decimales que puedan ser utilizadas por el programa en donde el algoritmo sea utilizado.

El caso switch on/off debe analizar y determinar si uno de los ángulos es obtuso en el triángulo a estudiar dentro del algoritmo, si los ángulos son agudos se aplica ley de senos y no se advierte otra casualidad o es uno u otro.

Ahora, el cálculo de los ángulos dentro de la imagen debe ser hecho en función del factor del ángulo sólido de visión, este es el ángulo de captura de la cámara y depende de factores como el lente, la distancia focal y el enfoque.

El enfoque distorsiona en muy poca cantidad el ángulo de captura por lo que se toma como despreciable su efecto frente a los demás factores.

El lente era un factor importante y sigue siendo en casos especiales. La variedad de lentes presentes en diferentes cámaras hace que los ángulos de visión sean muy diferentes dependiendo de las cámaras, por ejemplo: las cámaras de formatos especiales como los panorámicos tienen una proporción muy diferente en donde el ancho es mucho más grande que la altura de la imagen, por ello los lentes tienden a ser ovalados y sus ángulos de captura son calculados especialmente.

Para el presente caso voy a tomar como base los lentes estándar de cámaras convencionales tipo *pin hole* (agujero pequeño) como se presentó antes en el Capítulo 2 y por ello la geometría del lente tampoco va a ser un factor predominante en el cálculo mientras se mantenga las restricciones antes mencionadas.

La óptica (ciencia encargada del estudio de los lentes y sus efectos con la luz) hace poco análisis de las estructuras de cámaras de un solo lente con respecto al ángulo sólido, por ello este estudio.

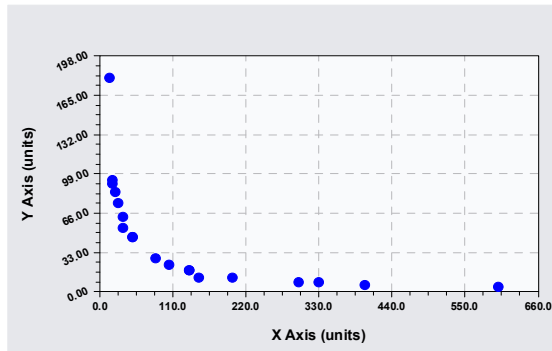
Temas relacionados se encontraron en manuales de cámaras como Cannon y Sony de lentes intercambiables. En las tablas relacionadas a los ángulos de captura y a las distancias focales de lentes con zoom variable se encontraron los siguientes datos expresados en una sola tabla ya que las distancias focales son diferentes para cada tabla en el manual de su respectivo lente:

	Distancia focal (mm)	Ángulo de Visión (grados)
1	15	179
2	20	91
3	20	94
4	24	84
5	28	75
6	35	54
7	35	63
8	50	46
9	50	46
10	85	28.5
11	105	23
12	135	18
13	135	18
14	150	12
15	200	12
16	300	8.25
17	330	8
18	400	6.16
19	600	4.25

Tabla:4.1 Distancia Focal vs Ángulo de Visión

Estos datos son utilizables a cualquier cámara incluidas las que se utilizarán en el prototipo demostrativo, de esta manera se puede incluir operativamente el dato de la distancia focal que es mucho más usado en la mayoría de las cámaras y no el ángulo de visión que es un dato escaso en la mayoría de las cámaras y lentes.

Ahora se muestra el gráfico “distancia focal vs ángulo de visión” en donde se ve claramente la figura de una hipérbola:



Y Axis= Ángulo de Visión (°)

X Axis= Distancia Focal (mm)

Figura: 4.3 Ploteo de puntos Distancia Focal vs Ángulo de Visión

Este gráfico fue obtenido del programa Curve Express el cual plotea las distribuciones de la tabla anterior colocando en las ordenadas el ángulo de visión y en las abscisas la distancia focal.

En la figura 4.3 se ve claramente una hipérbola que tenderá a cero al incrementarse la distancia focal y cruzará cerca de los 180° al acercarse a cero, se calcula la ecuación de la curva haciendo una extrapolación de datos y usando una regresión hiperbólica con ajuste matricial de tipo:

$$y = \frac{1}{a + x}$$

En donde Y sería el ángulo de visión y x la distancia focal, a es una constante.

Este ajuste es preciso pero se obtienen mejores resultados con el modelo de Harris, para esto la ecuación a encontrar sería:

$$y = \frac{1}{(a + bx^c)}$$

Con las mismas denominaciones adjuntando b y c como otras constantes.

Se puede tener un ajuste mucho más preciso, herramientas como Curve Expert (versión 1.3) pueden encontrar estos valores automáticamente y desplegarlos los datos directamente en forma visual, de esta manera se encontraron estos valores:

Harris Model: $y=1/(a+bx^c)$

Datos de coeficientes:

a = -0.053497604

b = 0.03048043

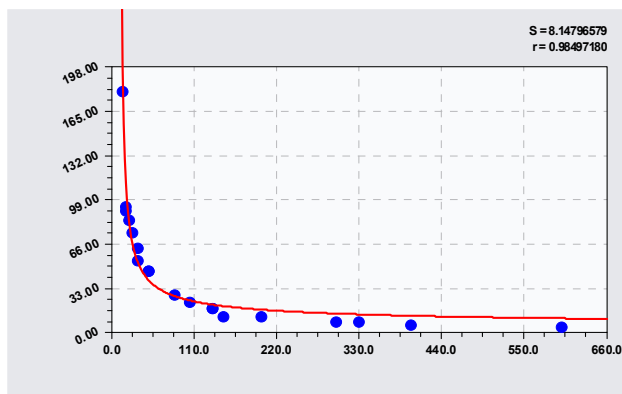
c = 0.24441286

Con estos datos se construye nuestra ecuación que relacionará la distancia focal con el ángulo de visión que sería reemplazando los valores de nuestras variables como se indicó antes:

$$AS = \frac{1}{(-0.053 + 0.030 \times DF^{0.244}} \quad (4.7)$$

Donde AS es el ángulo de visión y DF es la distancia focal.

Como se puede observar en la figura 4.4, se ha superpuesto la curva de la ecuación encontrada con los puntos distribuidos de la figura 4.3; la curva está muy aproximada a los datos mostrados en la distribución de la figura 4.3:



Ordenadas = Ángulo de Visión (°)

Abciscas = Distancia Focal (mm)

Figura: 4.4 Recta de ajuste sobre puntos ploteados

El programa Cuerve Express también indica el error residual de las muestras con respecto a los datos correspondidos en la ecuación encontrada, esto se observa en la figura: 4.5 y en su tabla:

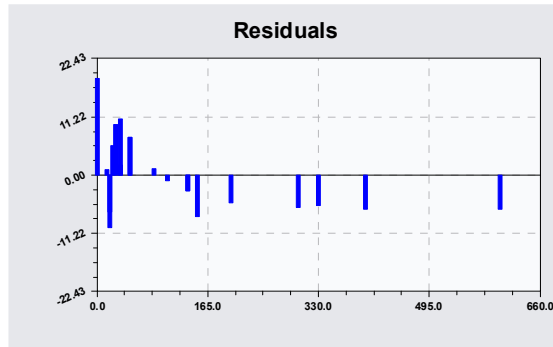


Figura: 4.5 Error residual tabulado

Harris Model: $y=1/(a+bx^c)$

Tabla residual:

Índice Distancia focal (mm)	Error de residuo
15	1.00689758
20	-10.10445816
20	-7.104458162
24	5.748685905
28	9.743108767
35	1.867165119
35	10.86716512
50	7.243365216
50	7.243365216
85	1.313747343
105	-1.056671567
135	-3.012391093
135	-3.012391093
150	-7.909232747
200	-5.305902938
300	-6.164043234
330	-5.836473755
400	-6.606913717
600	-6.612398701

Tabla: 4.2 Tabla residual

El residuo es el valor que se encuentra restando el punto encontrado por la ecuación y el punto fijado por la tabla 4.3.

El error residual llega a ser en su valor máximo apenas de 10 puntos sobre los 179 posibles, esto es claramente menor al 10% por lo que el resultado de la ecuación encontrada esta muy aproximada.

Si fuese necesario también se tienen los valores del historial chi cuadrado, entre ellos se denota los más importantes, en los puntos más críticos de la curva:

Harris Model: $y=1/(a+bx^c)$

Historial Chi cuadrado:

Iteración	Historial
It 3:	9.70442e+007
It 4:	1.40687e+007
It 7:	1.55316e+006
It 9:	4.8176e+006
It 11:	1.6433e+007
It 13:	7.0537e+006
It 15:	1.01599e+006

Tabla: 4.3 Historial Chi Cuadrado

El historial Chi Cuadrado es el valor cuadrático medio del valor del error de residuo de los valores próximos a todos los puntos ploteados con el objeto de encontrar un error mas representativo dentro de las inmediaciones de cada valor experimental, la iteraciones son la cantidad de valores adyacentes que se buscan al contorno del valor a buscar, así por ejemplo si con un punto se aplica una iteración 3 entonces se encuentra el valor cuadrático medio del error del residuo de el punto y un punto atrás y un punto al frente, si la iteración es 9 entonces se busca el error cuadrático medio del punto y cuatro puntos antes y después.

Como se ve, los puntos más críticos se encuentran en las iteraciones 3, 4 y 11 en donde el valor chi llega a ser extremadamente grande, esto se ve en la muestra de valores graficados en la figura: 4.3 en donde los valores de distancias focales pequeños muestran diferentes ángulos de visión que se alejan de una curva mas perfecta Este fenómeno se presenta en estas distancias focales debido a los sistemas más complejos de lentes que se necesitan para estos gran angulares mas conocidos como *ojo de pez* que deforman la percepción de la imagen.

Pese a que este fenómeno es observado no es necesario eliminar estos valores para encontrar una curva mas perfecta ya que para los siguientes valores el chi cuadrado es mucho más pequeño como se puede intuir de la gráfica del error residual denotando un acople mas preciso de la curva encontrada y los valores dato.

Una vez encontrado el ángulo sólido de visión se determina el ángulo en donde se encuentra el objeto a ubicar; para esto se necesita saber el ángulo que tiene el objeto con respecto al punto más izquierdo en la imagen. En la figura: 4.6 se puede observar lo dicho y como se encuentra el ángulo final que formarán los triángulos:

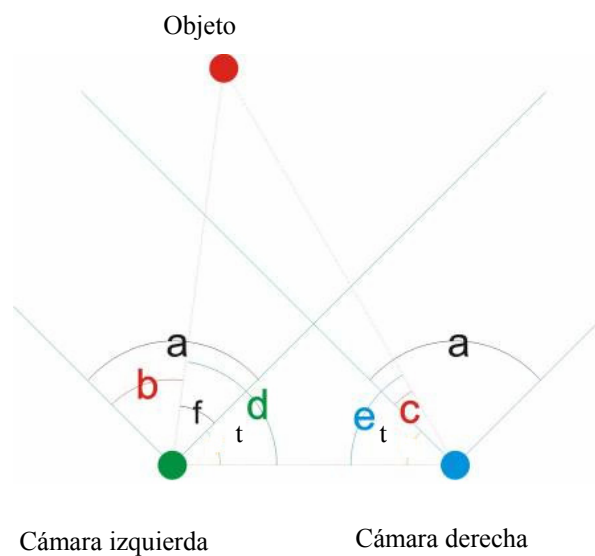


Figura: 4.6 Ángulos dentro de la triangulación simple

En la figura 4.6 se tienen las cámaras izquierda y derecha (puntos verde y azul respectivamente), sus ángulos de visión son los ángulos a , el objeto a encontrar es el punto superior (punto rojo), los ángulos b y c son los ángulos encontrados en las imágenes capturadas por cada cámara y se encuentra utilizando la distancia en píxeles del centro de masas del objeto al extremo izquierdo de cada imagen.

Los ángulos que se necesita para la triangulación son d y e ; y se logra encontrar de la siguiente manera:

$$\begin{aligned}
 t &= \frac{180 - a}{2} \\
 e &= r + c \\
 \alpha &= e
 \end{aligned}
 \tag{4.8a}$$

(4.8)

$$\begin{aligned}
 f &= a - b \\
 d &= r + f \\
 \beta &= d
 \end{aligned}
 \tag{4.8b}$$

Como se dijo antes, se precisa tener los ángulos b y c , para esto el algoritmo de ubicación de tramas en imágenes debe devolvernos esta posición en píxel utilizando un conteo de píxeles desde el borde izquierdo de la imagen hasta el centro de masas conocido, este dato es precisamente las coordenadas en píxeles del centro de masas, una vez adquirido este dato se procede a calcular los ángulos con tres observaciones principales:

La deformación angular de la lente es nula o casi nula (en la mayoría de los casos de lentes de cámaras, esto es cierto)

La imagen adquirida no ha sido escalada ni modificada en ninguna forma

Se conoce la definición de las cámaras en píxeles

Con estas observaciones se puede calcular el ángulo α y β de la siguiente manera y observando la figura: 4.7

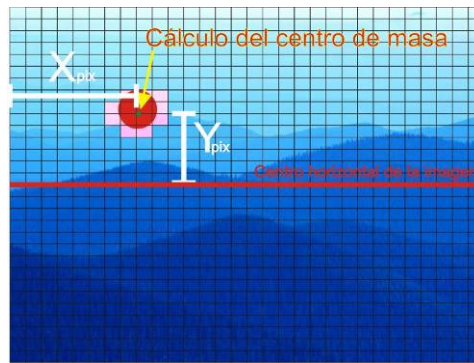


Figura: 4.7 Distancias para cálculo de ángulos en la imagen

$$\frac{\text{ángulo_sólido}}{\text{definición_horizontal}} = \text{defH}$$

$$\frac{\text{ángulo_sólido} * 3}{4} = \text{defV}$$

$$\angle(b_o_c) = X_{pix} * \text{defH}$$

$$\angle\text{elevación} = Y_{pix} * \text{defV}$$

(4.9)

Donde el ángulo b y c son los mismos de la figura: 4.6 , la definición horizontal y vertical son datos conocidos de las cámaras y definen la cantidad de píxeles que pueden entregar por cuadro de imagen; el ángulo de elevación es el segundo dato que necesita para las coordenadas polares de ubicación del objeto.

En este punto ya se obtiene los tres datos de las coordenadas polares (θ , elevación y r) donde, el primer dato (θ) es el ángulo “c” calculado con (4.8), el ángulo de elevación con (4.9) y la distancia radial r que se encuentra con la triangulación en (4.4).

Una vez encontrados los ángulos se prosigue a realizar la triangulación considerando todos los pasos que son:

- Tener la distancia focal de las cámaras y que estas sean de tipo *pin-hole* para que coincidan sus ángulos sólidos de visión con los encontrados por la ecuación de correspondencia (4.7).

- Las dos cámaras deben ver en paralelo, es decir, que sus normales sean equidistantes en todo momento.
- Tener una función dentro del lenguaje de programación cualquiera fuera este que nos devuelva la posición en píxel de la trama del objeto a encontrar.

4.3. EJEMPLO DEMOSTRATIVO:

Para efectos de demostrar este algoritmo en una prueba de escritorio utilizaremos imágenes simuladas como las que se muestran en las imágenes de la figura 4.8, supondremos cámaras que tienen una distancia focal de 40mm y una DIO de 30cm.

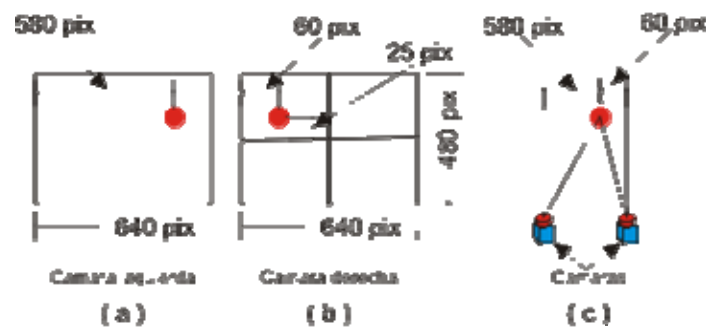


FIGURA: 4.8 Simulación de imágenes para un ejemplo de ángulos agudos

- (a) imagen de la cámara izquierda
- (b) imagen de la cámara derecha con eje horizontal y vertical en las normales
- (c) posicionamiento espacial de las cámaras y el objeto

En la figura 4.8 tenemos simuladas dos imágenes que tienen formatos de 640 x 480 píxeles, en la figura 4.8 (a) tenemos la imagen simulada de la cámara izquierda, la cantidad de píxeles que existen desde el borde izquierdo hasta el centro del objeto son 540, de esta imagen no necesitamos más información ya que con ella simplemente obtendremos el ángulo beta.

En la figura 4.8 (b) tenemos la imagen simulada de la cámara derecha, el objeto debe estar a la misma distancia de la horizontal en las dos cámaras debido a que estas están alineadas horizontalmente. En esta imagen se adquieren dos

datos, el primero es la distancia en píxeles desde el borde izquierdo que es de 60, la segunda es la distancia en píxeles del objeto a la recta horizonte que es de 25; el horizonte que es la línea media en la distancia vertical, es decir que esta línea está ubicada en los 240 píxeles verticalmente (la mitad de 480 píxeles que es la definición vertical). Igualmente la recta vertical se encuentra en la posición 370 píxeles (la mitad de 640 píxeles que es la definición horizontal) de la distancia horizontal de la imagen y es la recta normal a la cámara ubicada en la mitad horizontal, el punto centro del cruce del horizonte y la normal es la recta eje del radio en coordenadas polares correspondientes a los ángulos cero de θ y la elevación.

Con solo estos datos se puede encontrar al objeto espacialmente con referencia de la cámara derecha.

El primer paso es encontrar el ángulo de visión, para esto necesitamos el dato de la distancia focal, como dato esta era de 40mm, con ella utilizaremos la ecuación (4.7) que relaciona la distancia focal y el ángulo de visión:

$$\begin{aligned}
 AS &= \frac{1}{(-0.053 + 0.030 \times DF^{0.244})} \\
 AS &= \frac{1}{-0.053 + 0.030 \times (40^{0.244})} \\
 AS &= 48,089
 \end{aligned}
 \tag{4.10}$$

El ángulo sólido calculado es de 48.089 grados, en la práctica este ángulo puede ser tomado como 48.1 grados.

Con este ángulo y la definición de las imágenes podemos ahora calcular los ángulos α y β referentes en las figuras 4.1 y 4.2 y utilizando las ecuaciones (4.8^a) y (4.8^b):

$$\begin{aligned}
 t &= \frac{180 - 48.1}{2} \\
 t &= 65,95 \\
 c &= \frac{48.1}{640} * 60 \\
 c &= 4,50 \\
 \alpha &= 65.95 + 4.5 = 70,45 \\
 b &= \frac{48.1}{640} * 580 \\
 b &= 43,59 \\
 f &= 48.1 - 43,59 \\
 f &= 4,51 \\
 \beta &= 65.95 + 4.51 = 70,46
 \end{aligned}$$

Un vez encontrados los ángulos α y β podemos calcular la elevación, para ello utilizamos las relaciones de las imágenes que son de 4x3 como ya se explicó antes, con ello podemos calcular el ángulo vertical de visión:

$$\text{Angulo_vertical_de_visión} = avv = \frac{av}{4} * 3 = \frac{48.1}{4} * 3 = 36,075 \quad (4.11)$$

Con el avv (ángulo vertical de visión) podemos calcular la elevación del objeto con respecto al horizonte de la cámara derecha de la siguiente forma y utilizando las ecuaciones (4.9):

$$\begin{aligned}
 \text{elevación} &= \frac{avv}{480} * 25 = \frac{36.075}{480} * 25 \\
 \text{elevación} &= 1,87
 \end{aligned} \quad (4.12)$$

Con esto ya tenemos dos de los tres datos en coordenadas polares, θ y elevación son los dos datos que se necesitan para ubicar al objeto, el radio o distancia al objetivo se encuentran con la trigonometría antes explicada de la siguiente manera y utilizando la ecuación (4.4):

$$r = \frac{DIO \times \text{sen}(\beta)}{\text{sen}(\theta)}$$

$$\theta = 180 - 70.45 - 70.46$$

$$\theta = 39,09$$

$$r = \frac{30 * \text{sen}(70.45)}{\text{sen}(39.09)}$$

$$r = 44,83\text{cm}$$

Y con ello tenemos ubicado al objeto en las coordenadas 36.075° de θ , 1.87° de elevación y 44.83 centímetros de radio con respecto al foco de la cámara derecha.

Capítulo 5

5. DESARROLLO DEL ALGORITMO

5.1. El lenguaje

El algoritmo tiene un proceso sencillo el cual fue buscado con el propósito de aligerar los cálculos que tendrá que hacer el procesador o la unidad de procesamiento central.

Si bien estos cálculos son sencillos, se necesita una gran cantidad debido al flujo de información; si se ven estos flujos se observa que cada cálculo que se realice se lo tiene que hacer sobre cada captura de imagen que nos entregue la cámara ya que sería la menor cantidad de errores y el mejor rendimiento que se podría lograr.

Una cámara estándar en operación normal entrega 30 cuadros por segundo en formato NTSC (National Television System Comité) el cual es el más usado en los Estados Unidos, además del PAL (Phase Alternating Line) que predomina en Europa.

Si una cámara nos entrega 30 cuadros por segundo, el tiempo para cada cálculo es el de un treintavo ($1/30$) de segundo o 0.034 segundos, en este tiempo se deben realizar todos los cálculos del algoritmo incluida la transferencia de los datos o el despliegue de los mismo.

Para el desarrollo del algoritmo se escogerá un lenguaje de programación de “alto nivel” debido a los procesos complicados que se tienen para ubicar una trama dentro de una imagen, un lenguaje de alto nivel no dispone de comandos o acciones para simplificar el proceso de programación.

Los lenguajes de programación de alto nivel no tienen las librerías ni funciones probadas ni optimizadas para estas funciones, si bien estas librerías de funciones pueden ser encontradas para lenguajes como C# o C++, estas son pagadas y de precios elevados; lenguajes matemáticos como Python tienen librerías con procesos pesados (cientos de megas para descargar desde el Internet) e igualmente costosas; los lenguajes de programación de medio o bajo nivel quedan descartados por su extrema complejidad para el propósito de desarrollo de este proceso.

El lenguaje que se ha escogido de la gran gama de lenguajes para procesamiento de imágenes ha sido el paquete *LabView de National Instruments*® el cual consta de funciones y librerías suficientes para realizar todos los procesos necesarios para el desarrollo de este algoritmo; una ventaja más es que la Escuela Politécnica del Ejército tiene licencias estudiantiles para este paquete lo cual facilita el desarrollo de este algoritmo.

5.2. Las cámaras

Para las cámaras se ha escogido un par de la marca *Unibrain* modelo FIRE-i de conexión IEEE 1394 o *firewire*, esto garantiza poco procesamiento en la transmisión de datos, acoplamiento sencillo al PC, estructura simplificada tipo estrella²⁶ y facilitación del uso de las herramientas del lenguaje de programación ya que este usa mucho la trama de transmisión de datos de video vía IEEE 1394.

²⁶ <http://zone.ni.com/devzone/cda/tut/p/id/5929> Configuraciones estrella de buses típicos 29 de enero de 2008

5.3. El algoritmo

Un diagrama genérico del algoritmo se presenta en la figura 5.1:

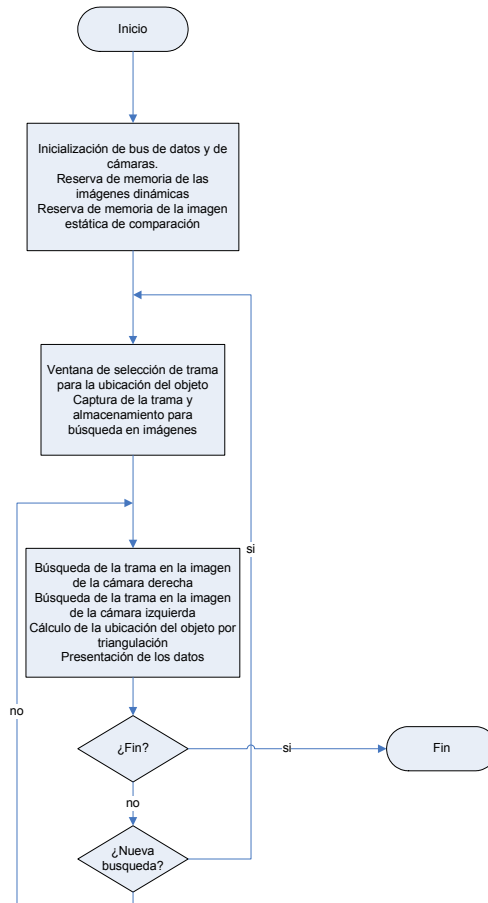


Diagrama 5.1 Diagrama de Flujo del Algoritmo

En el primer bloque están concentradas las inicializaciones de los dispositivos y de los bloques de memoria que se reservarán para los cálculos y procesos del algoritmo. En cualquier lenguaje de programación será necesario este proceso.

El segundo bloque está destinado a comenzar el ciclo ininterrumpido de la búsqueda de la trama que necesariamente debe comenzar con la especificación de la misma, luego esta se tiene que almacenar para las futuras búsquedas.

El siguiente bloque es en donde el algoritmo entra en funcionamiento; en él se busca la trama en cada una de las imágenes y con ello se hacen los cálculos para la ubicación del objeto en el espacio.

El bloque de pregunta *fin* determina si se debe o no salir del ciclo ininterrumpido.

El bloque de pregunta de nueva búsqueda permite salir del ciclo de búsqueda y regresa al bloque de inserción de la trama para la nueva búsqueda que se supone se tomará como un nuevo objeto.

5.4. Introducción al lenguaje:

LabVIEW®™ es una herramienta gráfica de test, control y diseño mediante la programación. El lenguaje que usa se llama lenguaje G.

Este programa fue creado por *National Instruments* (1976) para funcionar sobre máquinas MAC, salió al mercado por primera vez en 1986. Ahora está disponible para las plataformas Windows, UNIX, MAC y Linux y ha llegado a la versión 8.20 y 8.21 con soporte para Windows Vista.

Los programas hechos con LabVIEW se llaman VI (Virtual Instrument), lo que da una idea de su uso en origen, el control de instrumentos. El lema de LabVIEW es: "La potencia está en el Software". Entre sus objetivos están el reducir el tiempo de desarrollo de aplicaciones de todo tipo (no sólo en ámbitos de Test, Control y Diseño) y el permitir la entrada a la informática a programadores no expertos. Esto no significa que la empresa haga únicamente software, sino que busca combinar este software con todo tipo de hardware, tanto propio -tarjetas de adquisición de datos, PAC, Visión, y otro Hardware- como de terceras empresas.

Principales usos

Es usado principalmente por ingenieros y científicos para tareas como:

- Adquisición de datos
- Control de instrumentos
- Automatización industrial o PAC (Controlador de Automatización Programable)
- Diseño de control: prototipaje rápido y hardware-en-el-bucle (HIL)
- Diseño Embebido
- Domótica²⁷

El lenguaje G es de alto nivel cuando se utiliza sobre su interfaz gráfica de “drag and drop” (arrastrar y soltar << sistema de bloques que se enlazan entre si o introducen funciones en una pagina de proyecto para diseño y construcción de programas >>) sin embargo se tiene también una interfaz para programar en línea de código de un lenguaje de alto nivel, también permite la introducción de espacios tanto para lenguajes como C# y C++®, Java®, Cobol ® entre los más usados como bloques de activeX sobre máquina virtual.

Se Utilizará los bloques enlazados en su interfaz gráfica, no solo porque no es necesario llegar a un mayor nivel de programación debido a que las funciones de las que se dispone son suficientes, sino porque es extremadamente sencillo programar de esta forma.

5.5. El programa.

El diseño del programa el cual se ha apegado el diagrama de flujo anteriormente mostrado es el que se muestra en la figura 2.2:

²⁷ <http://es.wikipedia.org/wiki/LabVIEW> 29 de enero de 2008

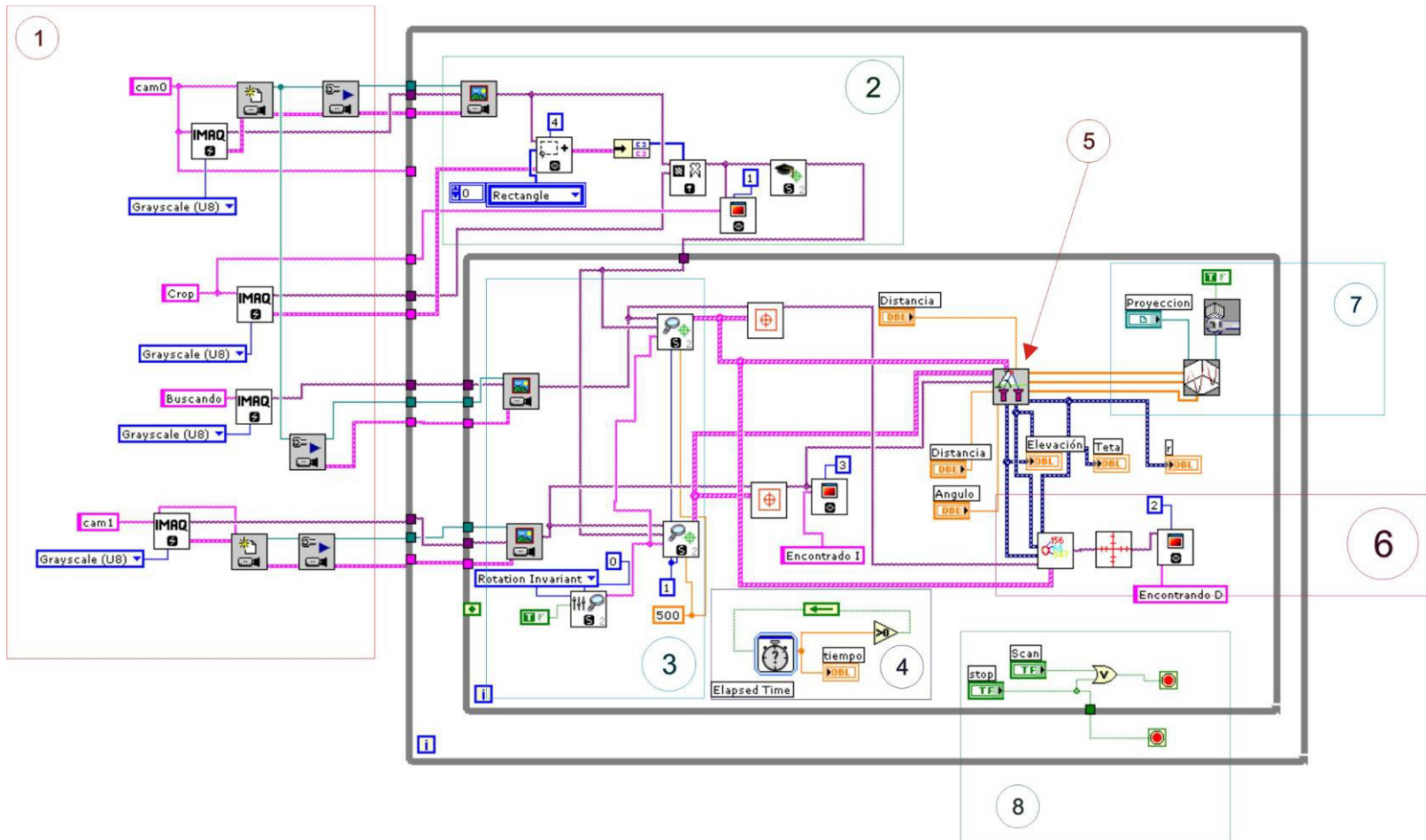


Figura: 5.1 Programación del algoritmo base en LabView

En la figura: 5.1 se muestran 8 bloques principales, en ellos están detallados los pasos del diagrama de flujo anteriormente expresado:

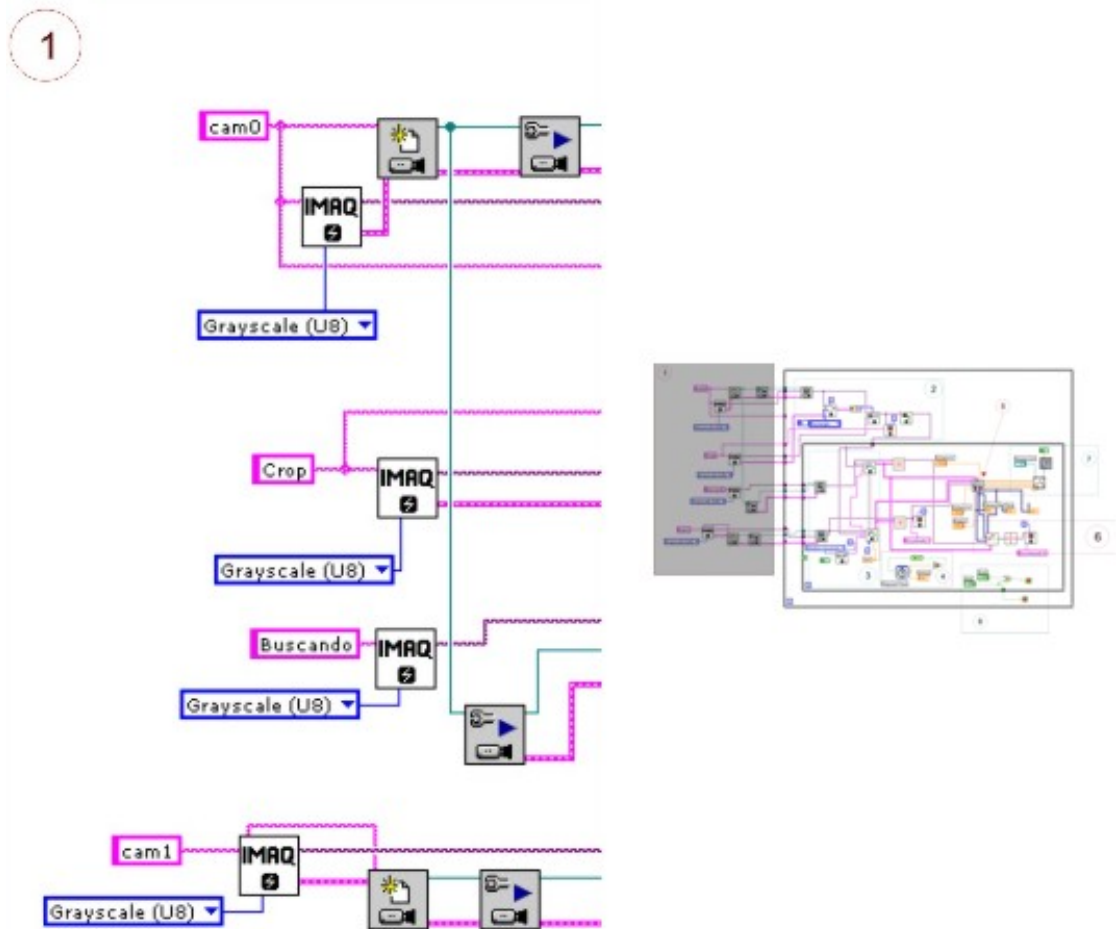


Figura 5.2: Bloque 1 Reserva de memorias y configuración de cámaras

1.- En este bloque se muestra la reserva de memoria (cam0, crop, buscando, cam1) y configuración de las cámaras firewire en donde se ve que se las configuró en 8 bits de escala de grises; también existe un bloque que inicializa la captura de imágenes de las cámaras y el envío continuo de imágenes hasta que el programa termine.

Con este bloque se realiza dentro del algoritmo mostrado en la figura 5.1 los pasos que inicializan el bus de datos y d cámaras, reservas de memorias dinámicas (cam0 y cam1) y la reserva de la memoria estática para la trama de comparación (crop y buscando).



IMAQ Create.- Es el bloque destinado a las reservas de memoria, los textos cam0, crop, buscando y cam1 son los nombres referenciales de estos espacios de memoria



IMAQ1394 Init.- habilita la transferencia de datos con las cámaras IEEE 1394, solicita las capacidades de la cámara, carga el archivo de las configuraciones de las cámaras.



IMAQ1394 Grab Setup.- Configura e inicializa el manejo de adquisición de datos con dispositivos IEEE 1394 ininterrumpidamente.

2.- Aquí están los parámetros necesarios para que se haga la introducción de la trama del objeto a buscar, en este bloque se realizan los pasos de ventana de selección de trama para la ubicación del objeto, captura de la trama y almacenamiento para búsqueda en imágenes, estos dos pasos son los comprendidos en el segundo bloque del algoritmo diagramado en la figura 5.1:

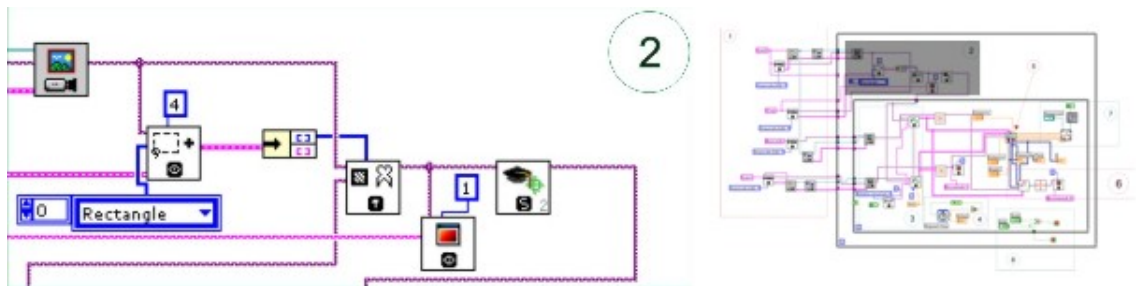
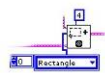


Figura 5.3: Bloque 2 selección de la trama a buscar



IMAQ ConstructROI.- El bloque de búsqueda de ROI (Region of Interest) que permitirá capturar la imagen de la cámara en ese instante y almacenarla en el espacio de memoria "Crop" además presenta la imagen en una ventana con identificador 1 y también entra en un bloque de aprendizaje de tramas.



IMAQ1394 Get Image.- adquiere el cuadro específico almacenado en el bufer del puerto IEEE 1394 correspondiente a la sesión del bloque IMAQ1394 Grab Setup previamente configurado



IMAQ Extract.- Recorta el espacio deseado de una imagen y lo almacena en un espacio de memoria (crop), con este recorte se piensa hacer las posteriores búsquedas de esta trama



IMAQ Learn Pattern 2.- Crea una descripción de la plantilla con la cual se piensa hacer la comparación en la fase de búsqueda.



El bloque de selección de ROI permite desplegar una ventana para la selección de la región de interés, esta ventana está con el subíndice 4 y se preseleccionó la selección de rectángulo; esto es necesario debido a que el corte de la imagen se realizó con un recortador de imágenes por coordenadas, el cual acepta dos puntos y con ellos realiza el corte rectangular de la imagen.

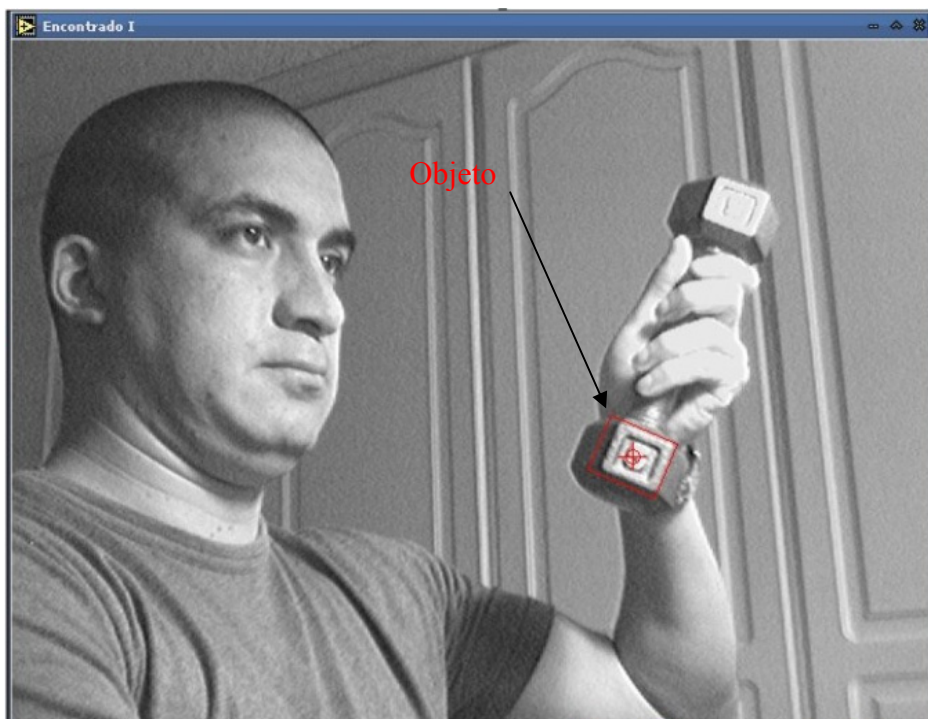


Figura 5.4 Ventana de selección de ROI

El bloque de selección de búsqueda comienza con el bloque de recorte, este se encarga de cortar la imagen para su presentación en la ventana con identificador 1 y su posterior aprendizaje en el bloque respectivo.

3.- Este bloque es el encargado de buscar la trama en las imágenes. En la primera parte están las capturas de la trama, luego el bloque que se encarga de programar la forma en la que la trama se buscará, en él está preseleccionado la rotación invariante lo que permite buscar la trama incluso si esta está girada; la calificación de la trama está en 500 en donde 1000 es una copia exacta de la trama y 0 es una trama completamente diferente a la buscada; la búsqueda se la realiza en las dos imágenes capturadas, tanto de la cámara derecha como la de la izquierda.

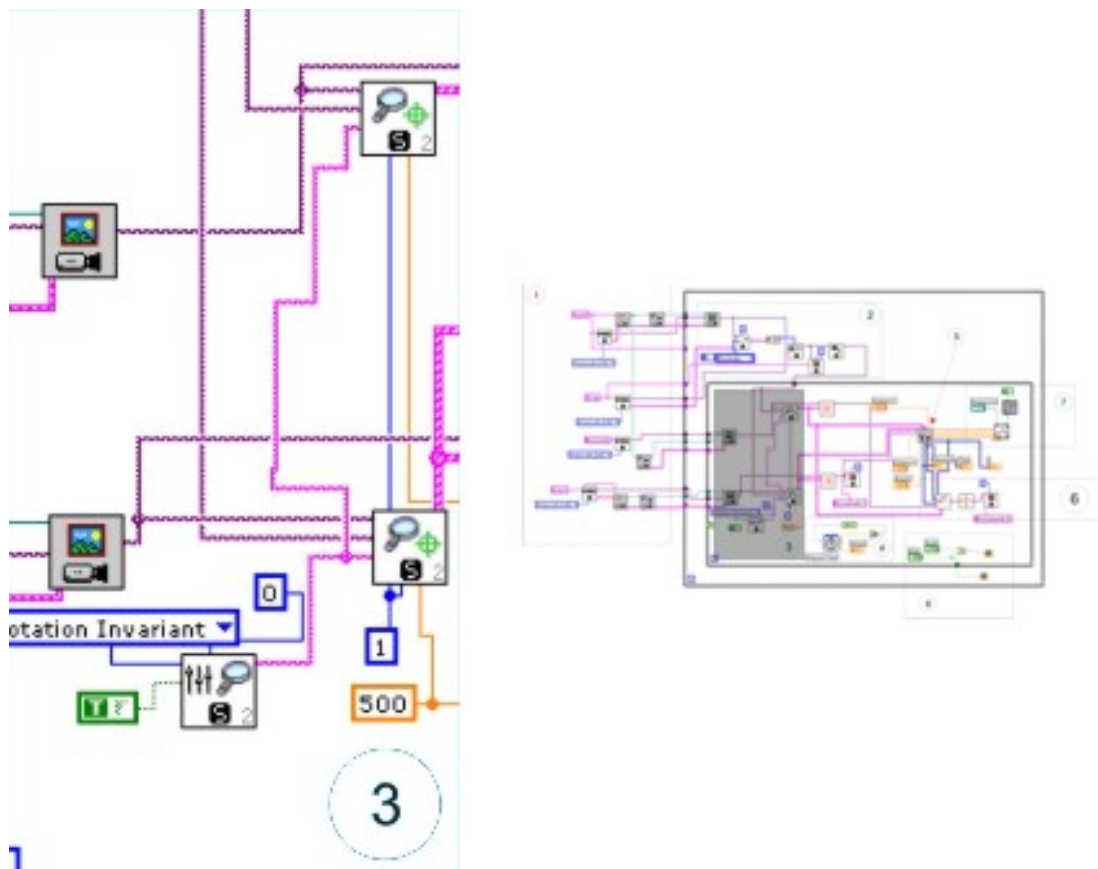


Figura 5.5: Bloque 3 Sección de búsqueda de tramas



IMAQ Setup Match Pattern 2.-Fija los parámetros para la fase de búsqueda, en el programa se seteo la rotación invariante con el objeto de hacer las búsquedas incluso si la trama esta en posición oblicua y la

búsqueda en subpíxeles para permitir pequeñas deformaciones de la trama provocadas por el paralaje.



IMAQ Match Pattern 2.- Realiza la búsqueda de una trama en una imagen a inspeccionar. Para este programa se calibraron los resultados en al menos 500 puntos, y se restringió a un solo resultado en la búsqueda, este bloque realiza una asignación de puntos con respecto a la diferencia que existe entre el histograma de la trama a buscar y la imagen a inspeccionar, de todos los posibles resultados el mejor se presenta debido a que se restringió a un solo resultado,

4.- Este bloque está incrustado para hacer el conteo del tiempo en el cual el algoritmo se desarrolla en la parte de la búsqueda, con ello se espera poder cuantificar el tiempo que toma buscar las imágenes y desplegar la información.

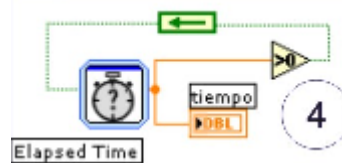


Figura 5.6: Bloque 4 timer de operaciones

5.- Contiene todo el algoritmo de búsqueda, posteriormente se expondrá lo que tiene este bloque ya que es el que contiene el algoritmo y es extensa su explicación.



Figura 5.7: Bloque 6 Presentación de resultados

6.- Permite la impresión de diferentes datos y arreglos dentro de la imagen mostrada con identificador 2 en donde se muestra el objeto encontrado, recortado y encerrado en un cuadrado para facilitar su búsqueda, también se dibuja una

cuadrícula de los grados espaciales en la imagen y 3 números, los cuales incluyen la altitud, la distancia y el θ con respecto a la cámara derecha.

Los dos bloques adicionales son construidos personalmente y contienen bloques que ubican textos, diagramas líneas y posicionan las mismas dentro de las imágenes como se muestran en la siguiente figura:

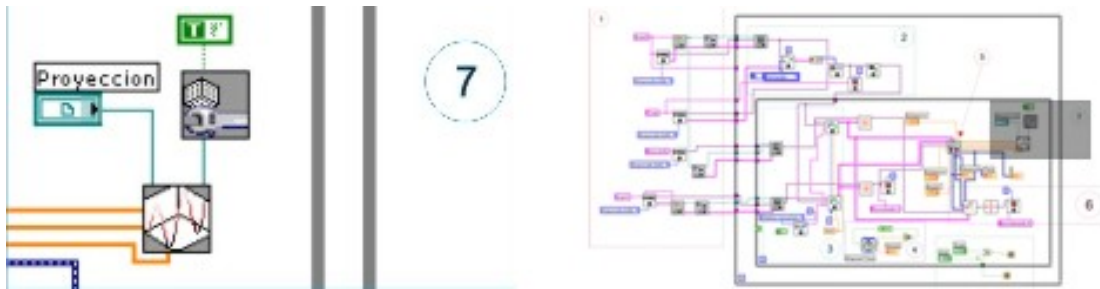


Figura 5.8: Bloque 7 ActiveX para representación en figura 3D

7.- Aquí están los ActiveX que permiten revelar el objeto en un espacio tridimensional simulado con objeto de referencia simple.

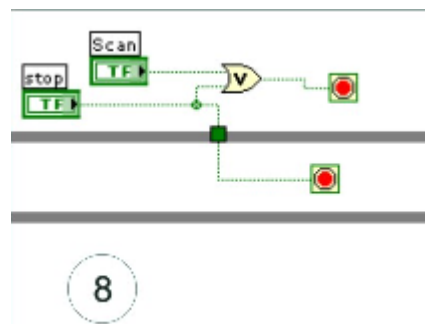


Figura 5.9: Bloque 8 Condiciones de fin de programa

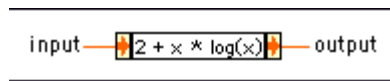
8.- Aquí están los dos bloques de salida del programa: el primero permite salir del bloque de bucle de búsqueda para entrar al bloque de inserción de la trama a buscar, el otro es el botón que permite salir del programa en general; solo existen cuatro reservas de memoria las cuales son borradas al finalizar el programa gracias a la nueva forma de trabajo de LabView 7.0 el cual determina los espacios de memoria usados y los borra al finalizar el programa automáticamente sin necesidad de llamar a la función de borrado de espacios de

memoria la cual era necesaria antes; otra de las razones es que los espacios de memoria son usados desde el inicio del programa hasta el final del mismo por lo que no hay necesidad de trabajar con la asignación de espacios y eliminación de los mismo en el curso del programa.

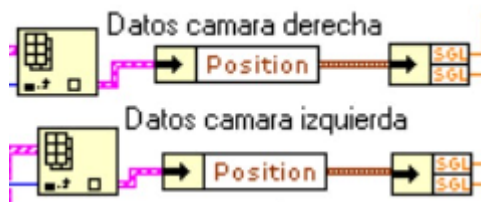
Esta asignación de memoria también es algo muy útil al momento de implementar en un dispositivo el algoritmo debido a que la necesidad de recursos tanto de procesamiento y de memoria no varía lo cual lo hace muy estable para ser preciso.

El algoritmo propiamente dicho está inmerso en el bloque que anteriormente se identificó, esto con el objeto de poder implementar el algoritmo a otros sistemas configurables por la necesidad de utilizar otro tipo de cámaras como las USB o videos composite.

El algoritmo se desarrolló como se ve en la figura: 5.10:



1^a.-Expression Node.- Recibe la distancia focal y se hace el cálculo del ángulo sólido. Este bloque es un bloque de operaciones que recibe un único dato de entrada y operándolo entrega una única salida, en nuestro caso aquí está la ecuación que relacionará la distancia focal y el ángulo de visión (4.7).



2^a.- Recibe toda la trama de datos proveniente de la ubicación de objetos y se segrega la posición solamente y las coordenadas de cada cámara, para ello usa 3 bloques:

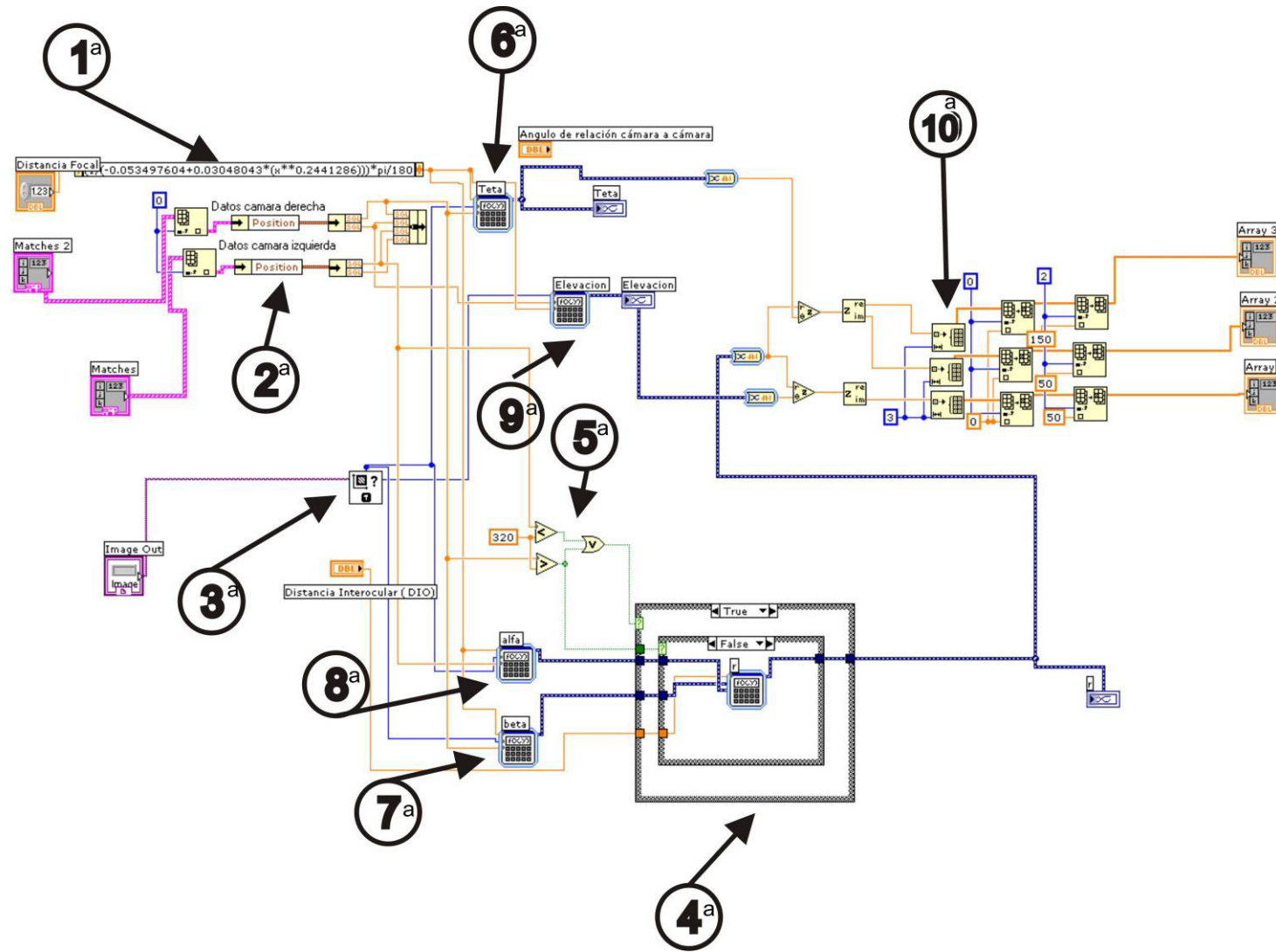


Figura: 5.10 Algoritmo programado en LabView



Index Array.- permite recuperar un dato aislado de un arreglo vectorial utilizando su índice.



Unbundle By Name.- aísla un dato de un grupo (cluster) por medio de su nombre



IMAQ GetImageSize.- Recibe la imagen y con ella solamente se obtiene el tamaño de la misma para la relación de distancias con respecto a los píxeles.

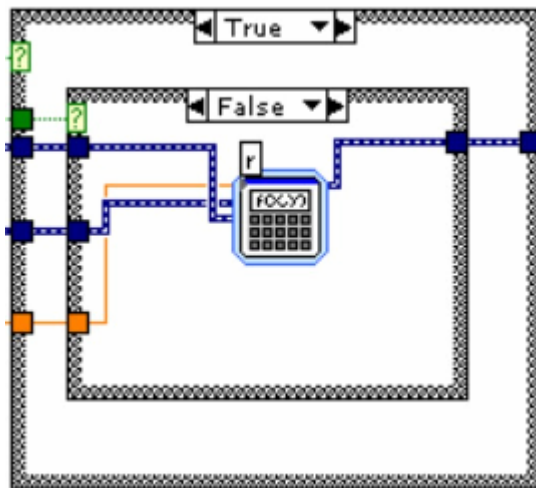


Figura 5.11: Switch on off para identificar triángulos agudos u obtusos

4^a.- Bloque Switch ON/OFF determina que parte del algoritmo utilizar, si el ángulo es mayor o menor a un ángulo recto para el cálculo de la distancia D o radio en coordenadas polares.



Este bloque permite las operaciones de múltiples entradas programadas como se deseen, se clasifican y etiquetan y luego se operan con ellas matemáticamente, estos pueden ser de cualquier tipo incluso permite operaciones entre valores de diferente tipo como enteros con booleanos.

En nuestro caso este bloque fue programado para el cálculo del radio, aquí se introducen tres datos, la distancia interocular denominada DIO, y α y β denominados alfa y beta respectivamente. En ellos se usó la fórmula 4.4 cambiando los valores a las nuevas denominaciones del bloque de esta manera:

$$\text{tag}(\beta) \times (\text{sen}(\alpha) \times \text{DIO}) + \cos(\alpha) \times \text{DIO}$$

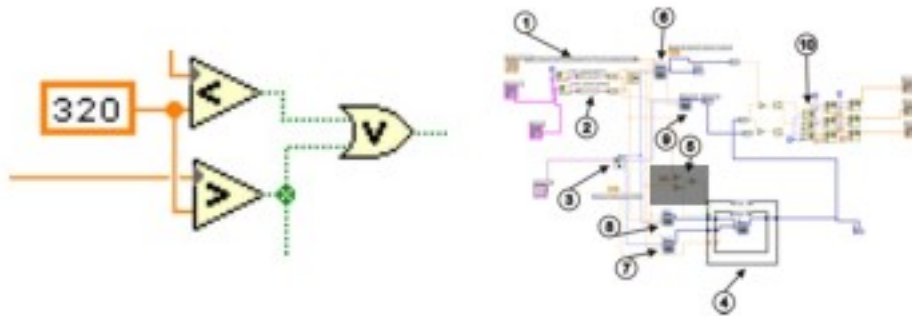
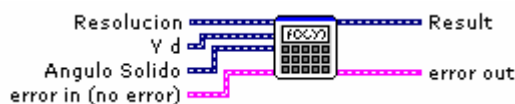


Figura 5.12: Determinación de ángulos mayores o menores a 90°

5ª.- Recibe los ángulos de las cámaras izquierda y derecha y determina si son mayores o menores a un ángulo recto (geométricamente no pueden ser los dos ángulos obtusos) se compara con 320 porque es el punto medio de la imagen en sentido horizontal, con esto se controla el bloque 4.



6ª.- Cálculo del ángulo de giro:

Se reciben 3 datos:

El ángulo sólido (Angulo)

La resolución de las cámaras (Resolución)

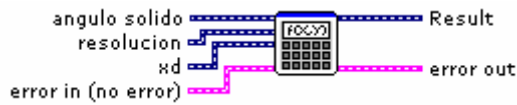
La distancia en píxeles de la cámara derecha desde el lado derecho (X)

$$\left(\frac{-\text{AnguloSólido}}{\text{Resolución}X} * \text{XDerecha} - \frac{\text{AnguloSólido}}{2} \right) * \frac{180}{\pi}$$

(5.1)

Esta ecuación se introduce de la siguiente manera:

$$-((\text{Angulo Sólido}/\text{Resolución } X)*X \text{ Derecha}-\text{Angulo Sólido}/2)*180/\pi$$



7ª .- Cálculo del ángulo beta (ángulo para la triangulación de la cámara derecha)

Se reciben 3 datos:

El ángulo sólido (ángulo)

La resolución de las cámaras (resolución)

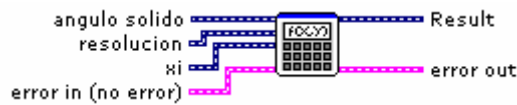
La distancia en píxeles de la cámara derecha (Xd)

$$\frac{\pi - \text{AnguloSólido}}{2} + \frac{\text{AnguloSolido}}{\text{resolución}} * xd$$

(5.2)

Igualmente esta ecuación se introduce de la siguiente forma:

$$(\pi - \text{ángulo sólido})/2 + ((\text{ángulo sólido})/\text{resolución}) * xd$$



8ª .- Cálculo del ángulo alfa (ángulo para la triangulación de la cámara izquierda)

Se reciben 3 datos:

El ángulo sólido (ángulo)

La resolución de las cámaras (resolución)

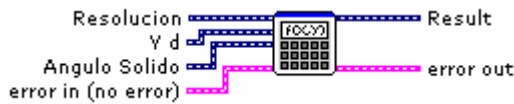
La distancia en píxeles de la cámara izquierda (Xi)

$$\frac{\pi - \text{AnguloSólido}}{2} + \frac{\text{AnguloSolido}}{\text{resolución}} * (\text{resolución} - xi)$$

(5.3)

Introduciendo en el bloque de esta manera:

$$(\pi - \text{ángulo sólido})/2 + ((\text{ángulo sólido})/\text{resolución}) * (\text{resolución} - xi)$$



9ª .- Cálculo de la elevación

Se reciben 3 datos:

La resolución de las cámaras (Resolución)

La distancia en píxeles de la cámara derecha con respecto a la base de la imagen (Yd)

El ángulo sólido de visión (ángulo sólido)

$$-\left(\frac{AnguloSólido * \frac{3}{4}}{Resolución} * Yd - \frac{AnguloSólido * \frac{3}{4}}{2} \right) * \frac{180}{\pi}$$

(5.4)

Ingresado en el bloque de la siguiente manera:

$$-((Angulo Sólido * 3/4) / Resolución) * Y d - (Angulo Sólido * 3/4 / 2) * 180 / \pi$$

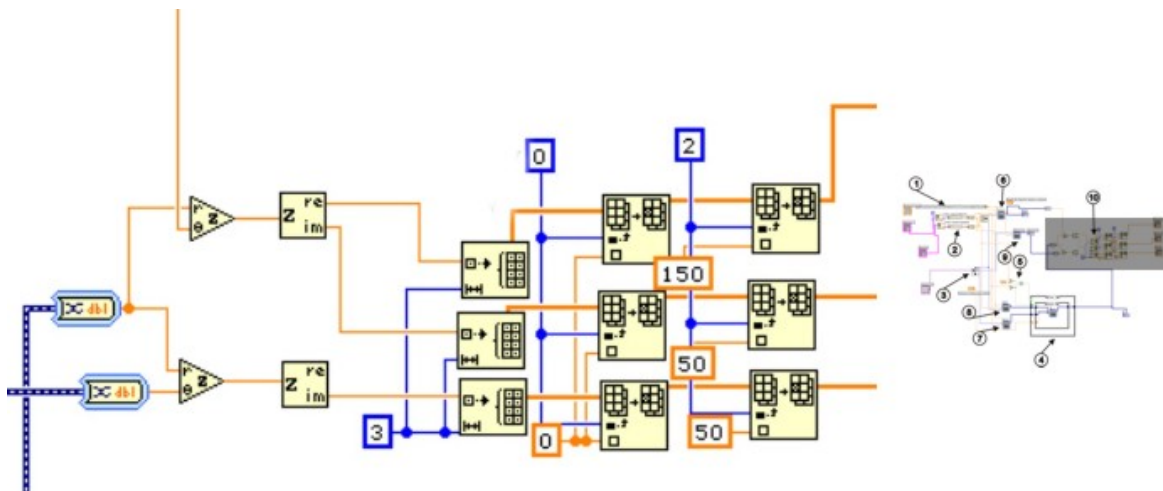


Figura 5.11: Separación de los datos y conversión a coordenadas rectangulares

10ª .- Separación de los datos que se encuentran dentro de tramas dinámicas para la presentación de los mismos en la imagen 3D, en el existe conversión de datos dinámicos a grupales, conversión de coordenadas polares a rectangulares y conversión de los datos en vectores.

Capítulo 6

6. Pruebas y resultados

Debido a que el algoritmo está presentado en un lenguaje de programación que tiene sus debidos límites al momento de calcular con decimales, se puede acotar que lo hace utilizando un flotante largo de 64 bits para alta precisión.

La cámara utilizada entrega tramas en definición estándar de 640 x 480 píxeles con 8bits de definición de grises (256 tonos configurados en LabView)

El cálculo del centro de la trama en donde se supone está el centro de masas del objeto se lo realiza utilizando las coordenadas de los puntos más externos del objeto y con la misma definición de 64 bits.

Por ello se puede tener una relación de error sobre la distancia y el tamaño de la trama como se observa en la figura: 6.1

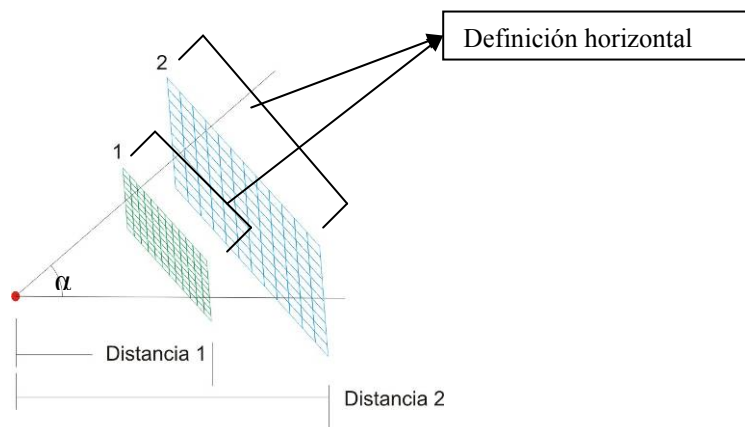


Figura: 6.1 Error sobre la distancia

Como se observa en la imagen 6.1, la trama se mantiene a diferentes distancias así que dependiendo de ella, el objeto a la cámara se tiene un error caracterizado por:

$$ef = \frac{As * r}{defH}$$

Donde “ef” es el error funcional, “As” es el ángulo sólido en radianes, “r” es el radio o distancia al objeto y “defH” es la definición horizontal de la cámara

Por ellos se sabe que a una distancia de 10 metros, y un objeto de 1 metro de ancho el error de su centro preciso se situará en un objeto que puede tener entre 1.01 y 0.98 m de ancho, y su centro se puede desplazar 0.006 m a lo ancho y 0.004 a lo alto, esto quiere decir que sobre el tamaño del objeto y la distancia, el error es menor del 0.1% teórico.

Estos cálculos son encontrados con el ángulo sólido en radianes de la cámara (42° obtenido de las hojas técnicas de la cámara) la distancia de 10m y la definición de 640 píxeles, lo que da un error de 0.01m por píxel de definición, esto sumando y restando del objeto de un metro y luego dividiendo este error para dos se tiene el desplazamiento posible del error para su precisión que como se indicó es del 0.1%.

El algoritmo tiene como fundamento principal ubicar objetos en el espacio, es decir que al momento en que las cámaras capturan las imágenes se puede tomar como el tiempo instantáneo en donde el objeto ha sido ubicado.

El algoritmo al ser implementado en un bucle se puede asumir como cálculos en “tiempo real”²⁸ ya que el tiempo que se demora en ubicar al objeto es menor a la percepción del hombre para identificar el proceso de cálculo y es menor al tiempo que demora la siguiente captura de imagen en ser procesada.

Estos cálculos en tiempo real permiten ubicar al objeto en “movimiento”; esto es debido a la incertidumbre con respecto a devolver los datos de la

²⁸ http://es.wikipedia.org/wiki/Tiempo_real 29 de enero de 2008

ubicación del objeto y el tiempo que demora en devolver un cuadro de imagen a la cámara; el error estaría relacionado con la velocidad del objeto.

Si el objeto tiene una velocidad instantánea “v” con aceleraciones instantáneas que generen incrementos menores al 10% en 1/30 de segundo de la velocidad, el tiempo que toma a las cámaras estándar capturar una nueva imagen es “t” y la distancia recorrida en este tiempo por el objeto es “s” tendríamos:

$$s = v * t$$

Es decir que el error que se puede presentar al ubicar un objeto debido a la velocidad del mismo está relacionado con la distancia que tiene este sobre el punto referente así podemos tabular las velocidades máximas a ciertas distancias que nos darán errores menores al 10 % que es un error aceptable dependiendo de las necesidades de exactitud y precisión. Si la distancia “r” es la distancia del objeto al punto referente y “e” es el error entonces:

$$\varepsilon = e * r$$

$$d < \varepsilon$$

$$v * t < \varepsilon$$

$$v < \frac{\varepsilon}{t}$$

$$v < \frac{e * r}{t}$$

Para un objeto que se encuentra a 1m de distancia al punto referente y para un error de 10% y un flujo de imágenes de 1/30s entre cuadros tendríamos:

$$v < \frac{0.1 * 1}{1/30}$$

$$v < 3m / s$$

La exactitud del algoritmo implementado está fuera de orden ya que lo importante dentro de un instrumento virtual es la precisión, la exactitud puede ser conseguida con calibración, no se podría observar un error de exactitud, la precisión está en función del error funcional y del error sistemático; por desgracia,

el lenguaje de programación no entrega en ninguna documentación ni un método para calcular el error sistemático de cálculos ni de acercamiento a la limpieza que haga en la imagen para determinar este error, por experiencia y comprobación física no se ha podido establecer un error debido a las dificultades para poder medir estas distancias con mayor precisión, se supone que este debe ser menor al 0.6% debido a los errores que se pueden encontrar en algoritmos similares y arquitecturas parecidas en otros sistemas. No se puede admitir un error mayor ya que la teoría no lo permite, la triangulación es exacta y precisa sin incluir ningún error.

Una prueba de escritorio se construyó de la siguiente manera tomando en cuenta el estado inicial de las cámaras y dispositivos de captura:

- Las dos cámaras fueron situadas a una distancia de 30cm una de otra (DIO), esta medida fue tomada con un flexómetro graduado en milímetros.
- Las dos cámaras usadas tienen la misma definición y ángulos de captura ya que son de la misma marca, fueron configuradas en proporciones 640x480 píxeles de definición, 8 bits de escala de grises y a 30 cuadros por segundo.
- Se utilizó una trama impresa en papel de 10x10cm con centro de masas conocido para la búsqueda.
- El paralelismo entre las normales de las cámaras se logró utilizando una regla de 30cm que fue posicionada a 10 metros del plano de las cámaras, así cada cámara ubicaba en su centro un extremo de la regla, la regla posee un clinómetro con objeto de asegurar la horizontalidad de las mediciones, también se situó a la misma altura de las cámaras para asegurar horizontalidad con respecto al suelo.
- Se utilizó un retazo de piola el cual se fijo tanto a la cámara derecha como a la trama con objeto de mantener fija la distancia fija y observar el error que se produce al momento de obtener el radio hacia el objeto como se muestra en la figura 6.2.

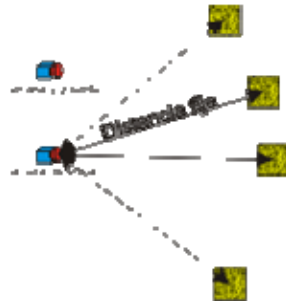


Figura 6.2: Prueba de escritorio

Fijando a diferentes distancias y realizando 5 mediciones para realizar un promedio cuyas mediciones supongan un 20% utilizando el prototipo y variando los ángulos desde cada borde y el centro de la imagen se obtuvieron los siguientes errores absolutos y su promedio:

Distancia de la piola (m)	1 (m)	2(m)	3(m)	4(m)	5(m)	promedio(m)
1.5	1.45	1.47	1.50	1.52	1.55	1.498
2	1.96	2.03	1.96	2.11	2.04	2.024
2.5	2.46	2.48	2.43	2.42	2.45	2.448
3	2.95	2.96	2.94	2.93	2.95	2.948
3.5	3.42	3.41	3.39	3.33	3.37	3.384
4	3.86	3.98	3.82	3.79	3.87	3.864
4.5	4.26	4.35	4.18	4.21	4.28	4,256
5	4.76	4.86	4.74	4.80	4.69	4.77

Tabla 6.2: Valores medidos por el algoritmo y promedios

Con estos datos podemos realizar una gráfica de las mediciones y sus porcentajes:

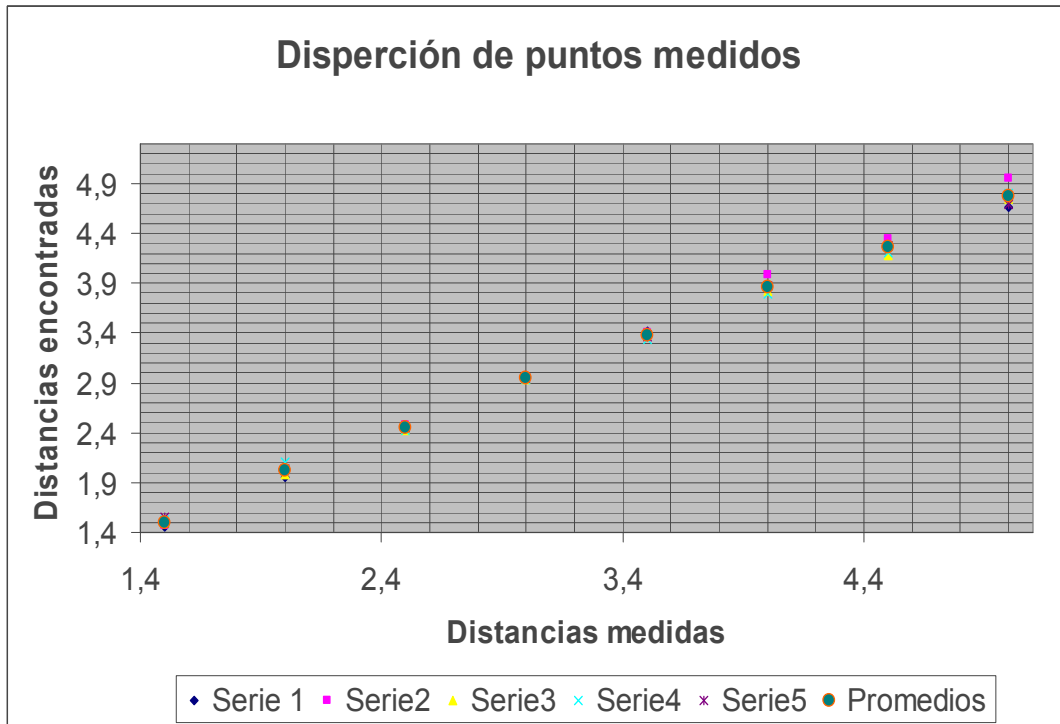
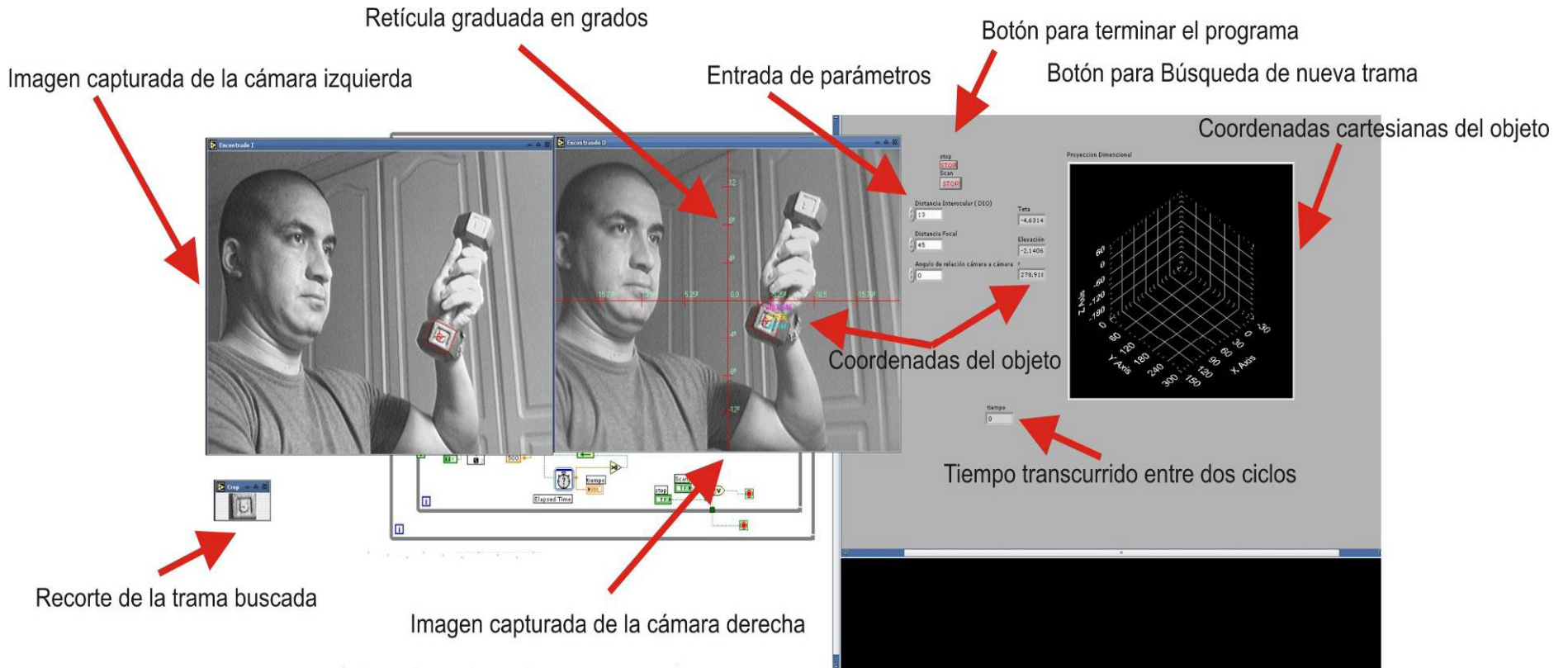


Figura 6.3: Dispersión de puntos medidos vs encontrados

La diferencia de los datos encontrados y los medidos son producto de:

- Errores al momento de encontrar el plano de las cámaras y del paralelismo de las normales de las cámaras; las cámaras utilizadas tienen las lentes sin alinear a ninguna parte de la estructura de la cámara como tal, por ello se diseñó el sistema antes explicado para determinar estos parámetros pero no son un método fiable, con cámaras más precisas se puede encontrar un mejor método para evitar estos errores.
- Errores por deformaciones de las lentes; las cámaras utilizadas tienen lentes plásticos de bajo costo, por ello se puede esperar deformaciones que no son especificadas en ningún documento de las cámaras que provocan fallas al momento de determinar los ángulos de objeto hacia las cámaras.
- Deformaciones visuales producto de la limpieza automática realizada por las cámaras; internamente las cámaras contienen bloques integrados para limpiar las imágenes de errores como destellos y errores de captura que deforman en pequeña cantidad los píxeles aledaños a los que necesitan procesamiento.

Figura 6.2



El computador en donde se realizaron estas pruebas tiene entre sus componente necesarios:

Procesador AMD 64X2 4200+ a 2.8GHz

Tarjetas de video Nvidia 6200 y 6100 en coprocesamiento paralelo sobre tarjeta madre

Tarjeta de adquisición IEEE 1394 a 800 Mbp

Sistema operativo Mandriva 2008 virtualizando windows unattended edition.

Paquete de programas LabView versión 7.1 *

Paquete de herramientas Visio 7.0 *

Paquete de utilitarios NIIMAQ 1394 2.0.1 *

Drivers OHCI para Mandriva 64bits 2008 de CARIOT versión 2.8

* No se posee licencia, funcionó sobre el tiempo de pruebas.

El tiempo de demora sobre esta configuración de computador (configuración 1) fue de un promedio de 0.015s con todas las características (impresión de imágenes, impresión de tramas, cálculos extras)

El tiempo de demora sobre esta configuración de computador (configuración 2) fue de un promedio de 0.008s solamente con impresión de tiempo (eliminando las impresiones de imágenes y cálculos de coordenadas cartesianas.

(configuración 3) Virtualizando un computador de procesador 1.8 Pentium IV a 32 Bits y con las mismas configuraciones y con todas las características se obtuvieron tiempos que en promedio marcó 0.028s

(configuración 4) Virtualizando el mismo computador y sin las características extras demoró 0.025s en promedio

(configuración 5) Virtualizando un computador con procesador más lento, los tiempos superan los 0.03s, esto es más del tiempo que se necesita para no

sobrepasar el flujo de imágenes de las cámaras que a 30FPS (cuadros por segundo) es de los mismos 0.033s

No se tienen los tiempos precisos sobre un computador que tuviera un sistema operativo Windows Xp en forma nativa.

(configuración 6) Una simulación corriendo el programa compilado como ActiveX y ejecutado sobre Basic (lenguaje de programación) sobre un computador INTEL Core 2 DUO a 2.3 GHz utilizando Windows XP en forma nativa denotó tiempos de 0.12s en promedio.

Configuraciones	1	2	3	4	5	6
1	0.015	0.008	0.280	0.025	0.030	0.12
2	0.015	0.008	0.280	0.025	0.030	0.12
3	0.015	0.008	0.279	0.024	0.032	0.12
4	0.016	0.008	0.280	0.026	0.030	0.12
5	0.015	0.008	0.280	0.025	0.031	0.12
6	0.015	0.008	0.280	0.025	0.03	0.12
7	0.015	0.009	0.280	0.025	0.03	0.12
8	0.015	0.008	0.280	0.025	0.03	0.12
9	0.015	0.008	0.279	0.025	0.03	0.12
10	0.015	0.008	0.280	0.024	0.03	0.12

Tabla 6.3 Tabulación de tiempos en 10 búsquedas de 10s.

Los tiempos fueron encontrados en promedios de tiempos de 10s encontrados cada dos ciclos, es decir que fueron tomados cada 1/15s lo que son 2 cuadros entregados por las cámaras, promediados automáticamente, como se ve claramente estos no varían, similarmente las medidas fueron igualmente precisas, en las 6 configuraciones tuvieron los mismos valores invariables y previamente tabulados.

Claramente se puede observar que los tiempos necesarios para calcular la posición del objeto están por debajo de los 0.033 segundos que se tienen para que el siguiente cuadro adquirido por la cámara se solape al proceso de cálculos.

En un lenguaje de programación más nativo para cálculos sobre matrices como Matlab o Python (sobre sistemas Linux) se podría llegar a menores tiempos e incluso podría correr sobre procesadores mucho más lentos.

Los datos demuestran una alta precisión lo que permite conseguir exactitud con una mayor calibración, también demuestran una velocidad considerable al permitir que el sistema deje libre a la planta para más procesos cualquiera sean estos.

No se ha podido entregar valores de precisión con otras configuraciones de imágenes ya que no se dispone de otras cámaras para experimentar con otra configuración.

Otras configuraciones de cámaras y resoluciones no han podido ser aplicadas al no poder tener las características dentro de la naturaleza del entorno físico para corroborar los datos devueltos por el prototipo.

La impresión de las imágenes se ven como muestra la figura: 6.2; allí se observa claramente que el objeto es encontrado incluso de forma oblicua, también se observa que se tiene despliegue de imágenes, cálculo de coordenadas cartesianas, despliegue de las mismas en una figura, cálculo del tiempo de demora, diagramación de retículas e impresión de las coordenadas en imagen, entre otros. Todos estos procesos demoran el proceso de funcionamiento provocado por la cantidad de cálculos necesarios dentro de un procesador para realizar estas operaciones.

Los resultados encontrados en la prueba de escritorio demuestran que la precisión del prototipo decrece conforme la distancia aumenta, extrapolando las

medidas de la dispersión encontrada en la figura 6.3 podemos observar que el error aumenta de acuerdo a la figura 6.4.



Figura 6.3: Extrapolación de errores de datos medidos

Por ello podemos asumir que llegando a los 5 metros hemos alcanzado el diez por ciento del error sobre la medida deseada, estos se puede reducir con una mejor calibración ya que la precisión no varía de la misma forma.

Si utilizamos la desviación cuadrática promediada de de los valores de la tabla 6.2 con los valores promediados podemos encontrar la tabla y la dispersión siguiente:

Distancias(m)	Errores cuadráticos medios
1,5	0,0304
2	0,0432
2,5	0,0184
3	0,0088
3,5	0,0272
4	0,0488
4,5	0,0488
5	0,088

Tabla 7.1 Distancia vs Error cuadrático medio

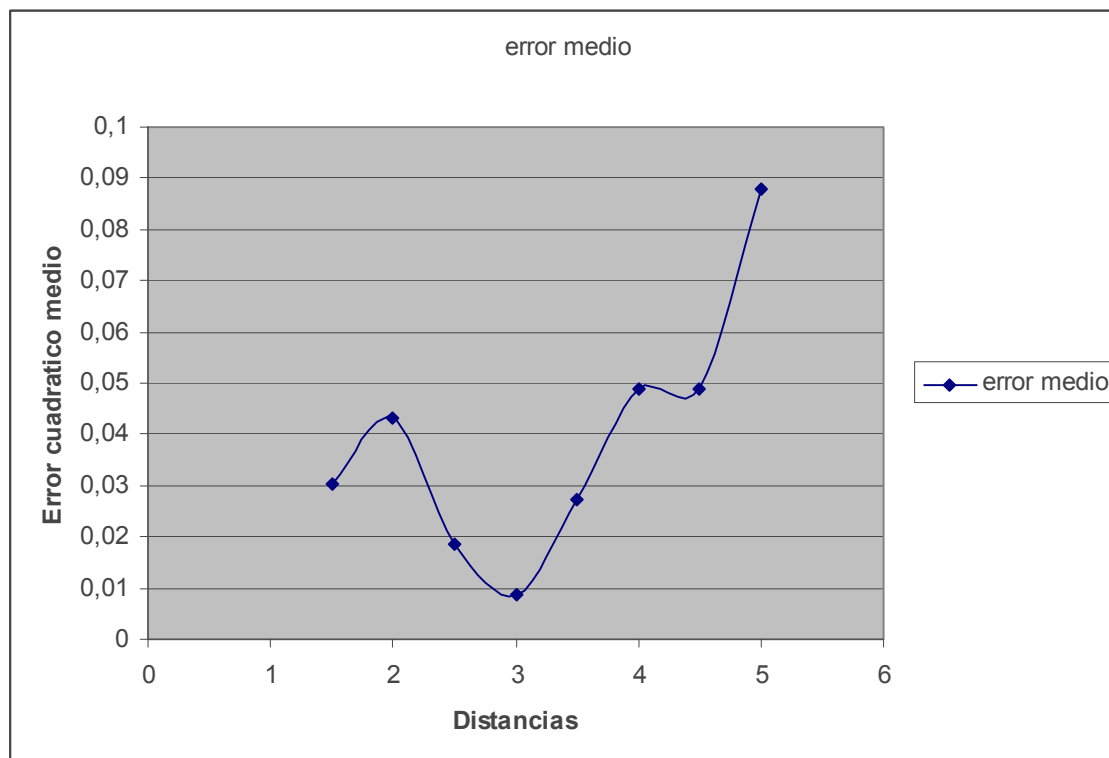


Figura 6.4: Dispersión de las distancias vs error medio

Con esta tabla y figura podemos observar como la precisión se pierde también en función de la distancia al objeto como se previó en el capítulo 6.

Con estos datos podemos determinar que el prototipo es funcional dentro de los 5m, fuera de ellos se espera que el error superará el 10%, el error de la desviación cuadrática en los valores cerca de los 3m bajan considerablemente pero esto se puede deber a simples coincidencias ya que en los demás valores se observa claramente como este error aumenta.

6.1. Comparaciones con otros métodos:

El método más usado para ubicar objetos de forma similar a la estereoscopia es el sonar, el cual tiene la desventaja de tener errores mayores y influencias más comunes al momento de encontrar el objeto; estas se deben a la longitud de la onda de choque, el medio en

el que se propaga, la capacidad de absorber la onda del objeto a encontrar, las ondas parásitas y de obstáculos parciales.

La estereoscopia en ventaja no tiene los mismos problemas, y los objetos que pueden afectar al momento de observar la trama son aquellos que tienen la misma o una trama similar en la imagen, por obvias razones un objeto que este obstruyendo la línea de vista también entorpece la búsqueda del objeto incluso si solo es parcial.

La posibilidad de que el objeto este camuflajeado en su entorno entorpece de gran manera al algoritmo.

Las limitaciones de la búsqueda de la trama están sujetos a los bloques de búsqueda del lenguaje que se usó para el prototipo, estos datos actualizados pueden ser encontrados y analizados en: <http://digital.ni.com/public.nsf/allkb/F05924D50AD6472B86256F320075CDBF> que es la página oficial de la información con respecto a este bloque de programación de la casa comercial del lenguaje y son continuamente modificados mientras el algoritmo se actualiza.

Capítulo 7

7. Conclusiones y Recomendaciones

7.1. Conclusiones

Todos los objetivos planteados en esta tesis fueron alcanzados. Si bien se introdujeron más bloques necesarios al momento de implementar el algoritmo, la velocidad y funcionabilidad del algoritmo es observada claramente.

La precisión de las medidas encontradas por el algoritmo está en función de 3 factores:

La definición de la cámara en pixeles.

El sistema antialiasing de la cámara al momento de presentar la imagen y los algoritmos de limpiado de la imagen del lenguaje de programación.

La cantidad de decimales considerados para el cálculo de la ubicación del objeto.

Los factores relativos a la velocidad y precisión fueron sometidos a pruebas que evidenciaron la ventaja del algoritmo de ubicación frente a otros métodos.

7.2. Recomendaciones

Se recomienda que el algoritmo investigado y desarrollado en este proyecto se implemente a un sistema básico y con un lenguaje de programación dedicado de bajo nivel, con el propósito de que se integren a sistemas y así ubicar un objeto en el espacio por medio de visión estereoscópica.

Se aconseja que para una siguiente prueba del algoritmo se integre un apuntador de cualquier tipo, sea este físico o luminoso con el objeto de observar la precisión del algoritmo al momento de integrarlo con otros sistemas.

Referencias Bibliográficas

- [1] *Visión por Computador* Luis Baumela <http://www.dia.fi.upm.es/~lbaumela/Alcala> Departamento de Inteligencia Artificial Universidad Politécnica de Madrid
- [2] <http://www.neoyet.com/vision.htm> VISIÓN EN ESTÉREO Juan José Samper Márquez
- [3] <http://es.wikipedia.org/wiki/Biónica> consultado al 29 de enero de 2008
- [4] [http://seneca.cucea.udg.mx/curso_linea/Sistemas%20Inteligentes/Modulo%201.%20Seccion%205%20\(1\).pdf](http://seneca.cucea.udg.mx/curso_linea/Sistemas%20Inteligentes/Modulo%201.%20Seccion%205%20(1).pdf) Antonio Reynoso Lobato 2002B CUCEA consultado al 29 de enero de 2008
- [5] http://es.wikipedia.org/wiki/Visión_artificial Visión artificial consultado al 29 de enero de 2008
- [6] *Visión por computador* Arturo de la Escalera Prentice Hall primera edición
- [7] <http://en.wikipedia.org/wiki/Stereoscopy> consultado al 29 de enero de 2008
- [8] <http://www.definicion.org/enfocar> consultado al 29 de enero de 2008
- [9] http://es.wikipedia.org/wiki/Distancia_focal consultado al 29 de enero de 2008
- [10] *Image processing with LabView and IMAQ Vision* Thomas Kilinger *Prentice Hall PTR*
- [11] <http://zone.ni.com/devzone/cda/tut/p/id/5929> Configuraciones estrella de buses típicos consultado al 29 de enero de 2008