



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

VISION ESTEREO Y ESTIMACION DE MOVIMIENTO.

T E S I S

QUE PARA OBTENER EL TITULO DE INGENIERO EN COMPUTACION

P R E S E N T A N :

JUAN JOSE JIMENEZ HERNANDEZ

ERIKA RODRIGUEZ GUZMAN

DIRECTOR DE TESIS: M.I. BENJAMIN VALERA OROZCO

SINODALES: DR. VICTOR GARCIA GARDUÑO

DR. BORIS ESCALANTE RAMIREZ

ING. LAURA SANDOVAL MONTAÑO

M. I. IGNACIO JUAREZ CAMPOS



MEXICO, D. F.

NOVIEMBRE 2000

285729



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Esta tesis está especialmente dedicada:

A mis padres por todo el apoyo brindado a lo largo de mi vida, porque siempre me motivaron y me impulsaron para seguir adelante.

A mi hermano por su cariño y lo importante que significa en mi vida.

A Ernesto, que me ha ofrecido su amor, su comprensión, su apoyo y su compañía que son tan importantes para mí.

A mi mejor amiga Elizabeth siempre juntas en las buenas y en las malas.

A mis buenos amigos Chucho, Huicho, Vic, Rodolfo, Miguel y Glico por todos los grandes momentos divertidos dentro y fuera de la escuela.

Al Ing. Joel Jaramillo, por su ejemplo, consejos y apoyo que me ha brindado.

A Benjamín por darme todo su apoyo y ser un buen tutor.

Erika

En primer lugar y antes que todo agradezco a Dios que en su misericordia me haya permitido concluir esta meta en mi vida. Desde que me permitió conocerlo de una forma especial, he sido testigo y participe de sus maravillas.

A mis padres, por el amor incondicional demostrado durante todo este tiempo, el apoyo moral, económico y social desde el comienzo de mi instrucción escolar, se ve coronado con la terminación de este trabajo, por lo mismo, a mi parecer satisface mas a ellos que a mí. A mis hermanos por la satisfacción de crecer y jugar juntos.

A la esposa y amiga que me acompañó durante el tiempo de estudiante en esta Universidad fue siempre inspiración y fuente de admiración en todo momento. A ella como ejemplo de fe constante que se necesita para terminar los objetivos, dedico este trabajo.

A todos los maestros que tuve durante toda mi vida de estudiante, y en especial a los maestros de esta facultad que me enseñaron no solo los conocimientos técnicos sino también los humanos. A la Universidad que me dio mucho.

A nuestro asesor campeón en paciencia y tolerancia, que sin sus sugerencias no se hubiera realizado este trabajo.

A los amigos que me acompañaron durante toda la travesía. En especial a mi compañera de tesis que con tenacidad y esfuerzo concluimos este sueño.

Hubiera yo desmayado,
sino creyese que tengo de ver la bondad
del Señor en la tierra de los vivientes.

Salmo 27:13

Juan José

ÍNDICE TEMÁTICO

Introducción	1
1. Conceptos básicos	7
1.1. Formación de secuencias de imágenes digitales	8
1.1.1. Video digital	8
1.1.2. Procesamiento de video digital	17
1.1.3. Formación geométrica de la imagen	19
1.1.4. Formación fotométrica de la imagen	23
1.2. Flujo óptico	24
1.2.1. Movimiento 2D vs. movimiento aparente	28
1.2.2. Estimación 2D del movimiento	20
1.3. Sistemas de visión estéreo	34
1.3.1. Aplicaciones de la estereoscopia	36
2. Estimación de movimiento basada en el gradiente	41
2.1. Modelo de movimiento 3D	42
2.2. Modelo de movimiento 2D paramétrico	46
2.2.1. Proyección del movimiento 3D en el plano imagen	46
2.2.2. Modelo de movimiento lineal simplificado	51
2.2.3. Modelo de movimiento constante	53
2.2.4. Modelos de movimiento 2D de más de dos parámetros	54
2.3. Minimización basada en el gradiente del modelo 2D de 4 parámetros	56

2.3.1. Diferencia del cuadro desplazado (DFD)	56
2.3.2. Optimización basada en el gradiente	57
2.3.3 Algoritmos basados en el método "Descenso a pasos"	60
3. Visión estéreo	71
3.1. Conceptos básicos	71
3.2. Calibración	82
3.3. Reconstrucción	86
3.4. Apareamiento estéreo	88
4. Programa para la estimación de movimiento EMMetro	89
4.1. Manual del operador	90
4.1.1. Información general	90
4.1.2. Descripción de EMMetro	91
4.2. Programación Visual Basic	98
4.2.1. Generalidades de Visual Basic	99
4.2.2. Control ActiveX DT Active Open Layers	102
4.3. Procedimiento típico para analizar secuencias de imágenes	110
5. Resultados y conclusiones	115
5.1. Resultados	115
5.1.1. Simulación en Matlab	118
5.1.2. Estimación de movimiento usando el programa EMMetro	121
5.1.3 Visión estéreo usando el programa EMMetro	128
5.2 Conclusiones y trabajo a futuro	133
Anexo A	135
Anexo B	137

Anexo C	141
----------------	------------

Bibliografía	151
---------------------	------------

INTRODUCCIÓN

Objetivo

Implementar técnicas de estimación de movimiento 2D basadas en visión estéreo y seguimiento de movimiento para mejorar el desempeño de un sistema de visión de dos vistas monoculares.

Definición del problema

Entorno actual

La visión estéreo habilita a los seres humanos y sistemas de visión a interactuar poderosamente con su entorno 3D, mediante el análisis de un par de imágenes (derecha e izquierda) conocidas como par estéreo. La técnica ha demostrado exactitud y economía en aplicaciones de visión por computadora, robótica y nivel industrial.

En la actualidad en el Centro de Instrumentos UNAM en el Laboratorio de Metrología, se ha desarrollado un sistema de visión estéreo a partir de imágenes fijas para mediciones tridimensionales.

Sin embargo, la inconveniencia del método empleado radica en las restricciones ambientales que se imponen para facilitar el procesamiento digital de las imágenes. En el trabajo previo [1], el

ambiente es forzado a tener una fuente de iluminación única que resalte el detalle a medir, como se ilustra en la figura 1. Para lograr tal situación, una fuente de luz láser apunta sobre el detalle a medir en el objeto bajo un ambiente carente de iluminación. De esta forma, las imágenes resultantes están formadas por el patrón de iluminación láser (casi un punto) sobre un fondo negro. Entonces la identificación de puntos tridimensionales a partir de dos imágenes simples (casi binarias) se vuelve una tarea sencilla de procesamiento digital de imágenes.

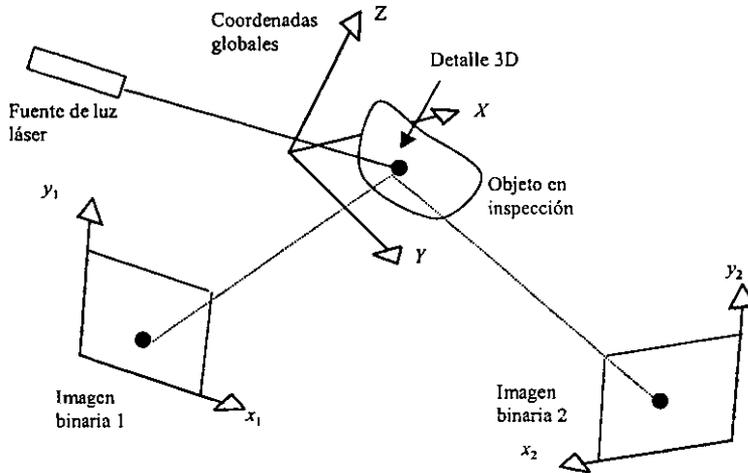


Figura 1. Visión estéreo a partir de imágenes fijas restringidas.

Descripción del problema a resolver

La inconveniencia del método descrito en el apartado anterior, sugiere el desarrollo de nuevas técnicas basadas en visión estereográfica y estimación de movimiento sobre imágenes en escalas de grises (no binarias), para que la estimación de objetos en el espacio resulte una alternativa confiable, ya que la tecnología actual en procesamiento de imágenes lo permite. Como parte del proceso, una computadora captura y procesa secuencias de imágenes rápidamente para estimar en forma robusta la situación de una región plana en el espacio. En el esquema de la figura 2, el par de imágenes en escala de grises es procesado mediante la técnica de estimación de movimiento para reconstruir la región espacial. La estimación de movimiento permite eliminar

las restricciones en iluminación mencionadas en el apartado anterior. Tal situación es planteada como una mejora al método desarrollado anteriormente en el Laboratorio de Metrología del CI UNAM.

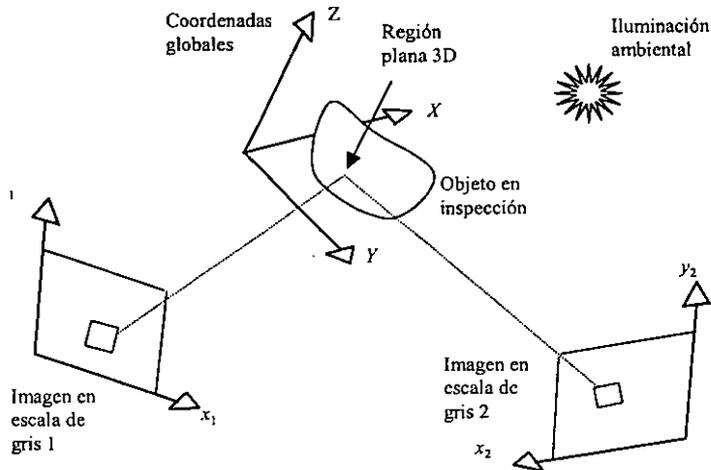


Figura 2. Visión estereoa a partir de imágenes fijas sin restricciones en iluminación.

Relevancia

El Centro de Instrumentos, en su Laboratorio de Metrología, contará con técnicas mejoradas para estimar dimensiones, utilizando visión por computadora, en forma robusta y sin restricciones en el tipo de imágenes. Lo anterior, facilita el proceso de medición por visión al eliminar la necesidad de condiciones ambientales especiales. Eventualmente podrá prestar los servicios de esta herramienta a la industria.

Relación con otras áreas

Existe una gran relación con las áreas de visión por computadora, metrología dimensional, procesamiento digital de imágenes, instrumentación, y programación de computadoras.

Método

El método involucrado en este proceso de medición de objetos en tres dimensiones es la utilización de las técnicas de visión estéreo, visión por computadora y estimación de movimiento para el procesamiento de un par de imágenes y la reconstrucción de posiciones tridimensionales. El esquema propuesto se ilustra en la figura 3. En tal figura, un programa en Visual Basic captura imágenes en tiempo real, procesa los algoritmos de estimación de movimiento, y proporciona la correspondencia entre pares de imágenes para su utilización en visión estéreo. Al contar con ésta herramienta de análisis de movimiento en secuencias de imágenes, los procesos de visión estéreo (calibración y reconstrucción) se benefician, y el proceso de medición se facilita.

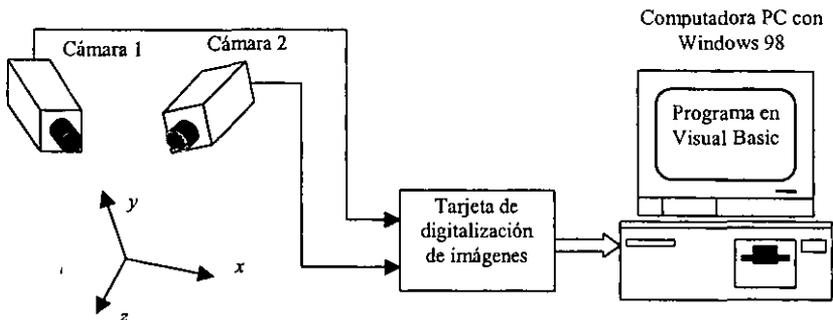


Figura 3. Método para la visión estéreo.

Resultados esperados

Implementar herramientas de visión por computadora y estimación de movimiento que operen con secuencias de imágenes capturadas en tiempo real, para la estimación robusta de posiciones espaciales con incertidumbres cercanas a un milésimo de la dimensión mayor.

Resumen de la tesis

El trabajo está dividido en 5 capítulos; para su mejor entendimiento se describen brevemente a continuación:

En el capítulo 1 se explicaran conceptos básicos de video digital y se abordaran fundamentos necesarios para comprender el entorno de trabajo relacionados con la geometría y fotometría de las imágenes y el concepto de flujo óptico necesario para entender el análisis del movimiento. Al final del capítulo mostramos algunos ejemplos de los actuales sistemas de visión estéreo y sus aplicaciones.

Durante el capítulo 2 desarrollaremos las herramientas necesarias para la estimación de movimiento a partir del concepto de gradiente. Se definen modelos de movimiento para 2D y 3D. Finalizamos este capítulo con el modelo que utilizaremos considerando 4 parámetros.

Mostramos, en el capítulo 3, la técnica de visión estéreo empleada. Con bases en la geometría, planteamos las distintas ecuaciones que nos ayudaran a elaborar el análisis de nuestro par estéreo.

Ya en el capítulo 4, empleamos todo el análisis en anteriores capítulos como base del sistema desarrollado. Mostramos las facetas del programa EMMetro para la captura y análisis de secuencias de imágenes. Parte del código esencial es identificado y mostrado para una visión general de sus desarrollo. Como parte final de este capítulo es presentado el procedimiento para analizar secuencias de imágenes con ayuda del sistema.

Concluimos con el capítulo 5 en el cuál mostramos los resultados de nuestra simulación en Matlab del algoritmo del gradiente y además mencionamos los resultados de un experimento con el sistema EMMetro. Comparamos los resultados obtenidos entre el programa y la máquina de coordenadas empleada como patrón de análisis. También hacemos una remembranza de los obstáculos encontrados durante la realización de este experimento y finalmente se proporcionaran las conclusiones.

El anexo A explica brevemente el concepto de interpolación.

El anexo B hace referencia, teóricamente, de la transformación de las matrices de rotación, este material es necesario para el mejor entendimiento del capítulo 2.

El anexo C es código en Matlab y Visual Basic del algoritmo del gradiente.

CAPÍTULO 1

Conceptos básicos

A continuación explicaremos algunos aspectos relacionados al video digital, cuando hablamos de video nos referimos a información ilustrada (visual), incluyendo imágenes estáticas e imágenes que varían en el tiempo. El video trata de representar el movimiento tridimensional en una proyección bidimensional, dando por resultado una imagen variante en el tiempo denotada matemáticamente por $I(x, y, t)$, es decir, una señal de tres variables independientes, dos espaciales y una temporal. El video digital muestra esta señal en tiempo y espacio, dando lugar a una serie de imágenes fijas, denominadas secuencias, cada una de las cuales esta representada, como un arreglo de pixeles, y cada pixel tiene un valor numérico que identifica su color para desplegarlo. En este capítulo introducimos estandares de representación de video digital; así como los problemas a los que nos enfrentamos para estimar el movimiento en 2D, y los algoritmos basados en la ecuación del flujo óptico. Se define lo que es movimiento 2D y movimiento aparente y la gran importancia en la distinción entre ellos. Además describimos estimación de movimiento en 2D, que es el primer paso hacia el análisis del movimiento 3D, que será estudiado en otro capítulo.

1.1 Formación de secuencias de imágenes digitales

Llamamos *imagen* a una función bidimensional de intensidad de luz $I(x,y)$, donde x y y representan coordenadas espaciales y el valor I en un punto (x,y) es proporcional al brillo (o nivel de gris) de la imagen en ese punto, ver figura 1.1. Una imagen digital es una imagen que se ha discretizado tanto en las coordenadas espaciales como en el brillo; una imagen digital se puede considerar como una matriz cuyos índices de fila y columna identifican un punto de la imagen y el valor del correspondiente elemento de la matriz indica el nivel de gris en ese punto. Los elementos de una distribución digital de este tipo se denominan *píxeles*.

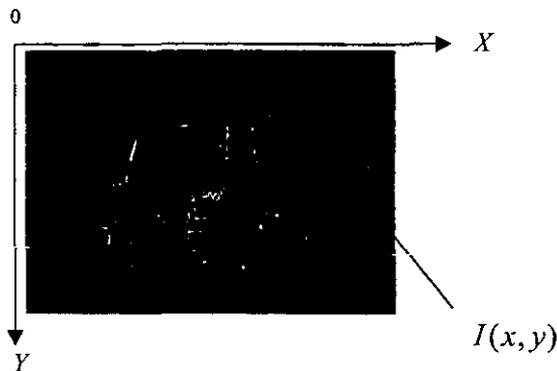


Figura 1.1. Convención de ejes utilizado para la representación digital de imágenes.

1.1.1 Video digital

En las dos últimas décadas se han dado avances importantes referentes a datos digitales y comunicación de voz. Primero hablaremos sobre las diferencias entre sistemas analógicos y digitales. En los sistemas analógicos la señal es interpretada por

una variación de voltaje, frecuencia o flujo magnético continuo en una cinta; por ejemplo en una grabadora, la información contenida en ella se encuentra distribuida de manera continua y sigue un orden espacio-temporal; para acceder la información determinada hay que recorrer el medio (cinta). No importa en qué punto de la longitud de la grabación examinemos, siempre encontraremos un valor para la señal grabada.

Estas son algunas de las características que debilitan el sistema analógico, ya que cualquier forma de onda es válida. Si la velocidad del medio no es constante, una forma de onda válida cambia a otra forma de onda también válida; así que no se pueden detectar errores en un sistema analógico. En conclusión, un error en voltaje simplemente cambia un voltaje válido en otro; en estos sistemas el ruido no puede ser separado. Una de las características de los sistemas analógicos es que las degradaciones no pueden ser separadas de la señal original, por lo que no puede existir ningún tipo de limpieza posterior.

Las señales de video son ondas eléctricas que permiten que nosotros capturemos imágenes en movimiento. Para poder capturar un objeto tridimensional en movimiento a través de un medio bidimensional, es necesario una exploración (scaneo), por ejemplo en las televisiones el scaneo consiste de rápidos barridos horizontales combinados con un barrido vertical más lento, así que la imagen es scaneada por líneas, este proceso es continuo. Este proceso convierte la resolución de la imagen en el dominio de la frecuencia. Si la imagen tiene una resolución alta se incrementará el número de líneas y aumenta el ancho de banda.

Uno de los principales tipos de color en el video analógico es el RGB (red, green, blue), las cuales son individualmente digitalizadas y emergen de los sensores de una cámara, necesitando un ancho de banda completo, este formato se usa en imágenes estáticas. En cambio con el video digital, no se necesitan sincronizar los pulsos, una computadora conoce exactamente donde empieza una línea así como el número de píxeles por línea.

Una forma de onda de video puede ser representada digitalmente con un sistema conocido como Modulación de Código de Pulsos (PCM) que es prácticamente de uso universal. PCM toma al tiempo en forma discreta, en vez de ser continuo, se va evaluando intervalos regulares a este proceso se le llama

muestreo y cuantización, y la frecuencia con la que se lleva a cabo se llama frecuencia de muestreo (f_s).

En los sistemas de televisión, la imagen que entra en el sensor de la cámara será continua en tiempo y continua en dos dimensiones espaciales, correspondientes al alto y ancho del sensor. En los sistemas de video analógicos, el eje horizontal es muestreado con cuadros, y el eje vertical es muestreado en líneas. El video digital simplemente agrega un tercer proceso de muestreo entre líneas.

La resolución horizontal y vertical en el video digital está relacionada con el número de píxeles por línea y el número de líneas por cuadro.

Una imagen monocromática digital sería un arreglo rectangular de puntos en los cuales se guarda un número correspondiente a la brillantez en él. Los puntos se conocen como elementos de imagen (picture elements), generalmente abreviados como pels o píxeles. Con la colocación cercana de los píxeles se espera que el observador perciba una imagen continua. Obviamente, entre más fino sea el espaciamiento entre píxeles, más grande será la resolución de la imagen y mayor cantidad de información se necesitará para almacenar la imagen, con ello aumentará también el costo del proceso.

Los arreglos de píxeles y líneas en una región contigua de la memoria se llaman bitmap. Hay cinco características importantes de un bitmap: el comienzo de la dirección en memoria, el número de píxeles por línea, el valor del campo, el número de líneas y el número de bits por píxel. El valor del campo especifica la distancia en memoria desde el comienzo de una línea hasta la siguiente, ésta formara un "cuadro-compuesto" en una región contigua de la memoria después de dos scaneos verticales. Cada señal usualmente se representa con 8 bits por píxel para evitar "contornos artificiales". Contorno significa una variación lenta de regiones de intensidad de la imagen debido a la insuficiente resolución del bit [2].

Estándares de video digital

Para el intercambio de video digital entre diferentes aplicaciones y productos se requieren formatos estándar de video

digital. Los datos del video necesitan ser comprimidos por lo cual se busca una compresión estándar. El video digital liga y acerca a tres campos de la tecnología: la industria de la computación con estándares de resolución en los despliegues, la industria de la televisión con estándares de estudios digitales, y en las telecomunicaciones con protocolos de comunicación estándares. Esta sección brevemente introduce alguno de estos estándares.

Número de pels/línea	720	720	360
	360	360	180
Entrelazado	2:1	2:1	1:1
Tasa temporal	60	50	30
Relación de aspecto	4:3	4:3	4:3

Tabla 1.1. Estándares del estudio de video digital [2].

El video digital no es nuevo en la transmisión de TV, donde la edición y los efectos especiales son representados por video digital ya que es muy fácil manipular imágenes digitales. Al trabajar con video digital también se evita ruido que sería de otra manera causado por grabaciones repetidas de video analógico en cintas durante varias escenas de producción. Otra aplicación para la digitalización del video analógico es la conversión entre diferentes estándares análogos, tales como PAL para NTSC.

La antigua CCITT (Comité Consultivo Internacional para el Teléfono y Telégrafo) propuso un nuevo formato para video digital, llamado el *Common Intermediate Format* (CIF). Los parámetros del formato CIF son mostrados en la tabla 1.1. Los formatos CIF son progresivos (no entrelazado), y requiere aproximadamente 37 Mbps.

En la industria de la computación, los estándares de resoluciones de video son establecidos por el Video Electronics Standards Association (VESA). Los estándares de la más vieja computadora personal (PC) son el VGA con 640 píxeles/línea x 480 líneas, y TARGA con 512 píxeles/línea x 480 líneas. Muchas resoluciones altas para estaciones de trabajo conforman el estándar S-VGA, el cual soporta dos modos principales, 1280 píxeles/línea x 1024 líneas o 1024 píxeles / línea x 768 líneas. La tasa de despliegue de estos modelos es de 72 cuadros/seg. Se han sometido múltiples propuestas a la Comisión Federal de Comunicaciones (FCC) para una alta definición de estándares de TV. Aunque tales estándares aún no han sido formalmente aprobados.

Hay varias aplicaciones de video digital, por ejemplo: alta definición HDVT, servicios multimedia, videoconferencia, y videoteléfono, que tienen diferente resolución espacio-temporal. Algunas de las opciones disponibles en redes y su tasa de bit son mostradas en la tabla 1.2.

Teléfono convencional	0.3-56kbps
Unidad fundamental BW de teléfono (DS-0)	56 kbps
ISDN (Integrated Services Digital Network)	64-144kbps (px64)
LAN (Local Area Network)	30 kbps
T-1	1.5 Mbps
Ethernet (Packet-Based LAN)	10 Mbps
Ancho de banda ISDN	100-200 Mbps

Tabla 1.2. Algunos protocolos y sus tasas [2].

La investigación y los altos requerimientos de las tasas de bit en redes, indica la viabilidad del video digital, está depende de cómo se podrían comprimir imágenes en video. Afortunadamente,

se ha observado que la calidad de reconstruir imágenes CCIR 601 después de la compresión por un factor de 100, es comparable a la calidad de video cinta analógica (VHS). La compresión de video es una tecnología importante para el desarrollo de varios productos de video digital, los estándares de compresión de video han sido desarrollado para varias tasas de transmisión y se siguen invirtiendo esfuerzos en la disminución de tales tasas de transmisión. Los principales estándares mundiales de la compresión de imágenes y video son listados a continuación.

CCITT G3/G4	Imágenes binarias (no adaptivo)
JBIG	Imágenes binarias
JPEG	Frame estático en escala gris y color
H.261	p x 64 kbps.
MPEG-1	1.5 Mbps
MPEG-2	10-20 Mbps
MPEG-4	4.8-32 kbps

Tabla 1.3. Estándares mundiales para la compresión de imágenes [2].

Los códigos CCITT del grupo 3 y 4 han sido desarrollados para la transmisión por fax y son actualmente usados en todas las máquinas de faxes. JBIG ha sido desarrollado para arreglar algunos de los problemas con los códigos CCITT del Grupo 3 y 4, principalmente en la transmisión de imágenes de medio tono. JPEG es un estándar de compresión de imágenes estáticas (monocromática y de color), pero también encuentra aplicación en compresión de video, más que nada por su gran disponibilidad en hardware VLSI. La recomendación CCITT H.261, está relacionada con la compresión de video para aplicaciones de videoconferencia sobre líneas ISDN. Típicamente la videoconferencia usa el formato CIF, que requiere 384 kbps, el cual corresponde a $p=6$. El

MPEG-1 maneja 1.5 Mbps para almacenamiento en formato digital CIF en CD-ROM y en disco duro. MPEG-2 es desarrollado para la compresión de alta definición de video en 10-20 Mbps con HDTV como una de las posibles aplicaciones.

La funcionalidad de varios productos de video digital requiere no sólo de estandarización del método de compresión sino también la representación (formato) de los datos. Hay una abundancia de formatos/estándar de video digital, además de los CCITT 601 y CIF estándar. Algunas propiedades de otros formatos estándar propietarios son mostradas en la tabla 1.4.

DVI (Video Digital Interactivo), Indeo	Corporación Intel
QuickTime	Computadoras Apple
CD-I(Disco Compacto Interactivo)	Consumidor Electrónica Phillips
Foto CD	Compañía Kodak Eastman.
CDTV	Commdore Electronics

Tabla 1.4. Ejemplos de formato estándar de video propietario [2].

¿Por qué video digital?

En el mundo de video analógico, nosotros tratamos con aparatos de TV, grabadores de videocasete (VCR) y videocámaras. Para la distribución de video contamos con las transmisoras de TV y con compañías de TV por cable que transmiten programación predeterminada a una tasa fija. El video analógico, debido a su naturaleza, proporciona una cantidad muy limitada de interactividad, por ejemplo, la selección de un sólo canal en la TV, la búsqueda rápido-delantera y la repetición lenta del movimiento en la VCR. Además, nosotros tenemos que vivir con el formato NTSC. Todo el video que se captura en un disco láser o cinta tiene que ser NTSC con sus bien conocidas propiedades y su calidad de imagen de cuadro estático muy baja. Para desplegar

señales NTSC en monitores de computadora o la TV europea, nosotros necesitamos transcoders caros. Para desplegar una pequeña versión del cuadro NTSC en una esquina del monitor, nosotros necesitamos reconstruir primero el cuadro entero y entonces digitalmente reducir su tamaño.

Nuevos desarrollos en tecnología y hardware de imagen digital están siendo implementados conjuntamente en la TV, la computadora, y las comunicaciones. La aparición de mejores algoritmos de compresión de video, redes de fibra óptica, computadoras más rápidas, tarjetas de video dedicadas, y la grabación digital prometen una gran variedad de productos de video digital y de comunicación visual. Hay aplicaciones comerciales tales como:

- HDVT.

Para 20 Mbps sobre canales de 6MHz.

- Multimedia, video de escritorio.

Para 1.5 Mbps en CD-ROM o almacenamiento en disco duro.

- Videoconferencia.

Para 384 kbps utilizando p x 64 kbps en canales ISDN.

- Videoteléfono y las comunicaciones de imágenes móviles.

Para 10 kbps que utiliza la red telefónica de cobre.

Otros ejemplos de aplicaciones son: procesamiento de imágenes como una medida de vigilancia para el ejército o aplicación de la ley, sistemas de carretera con vehículos inteligentes, mando de tráfico en puertos, en el cine (producción de películas), en la medicina, la aviación y simulación de mando de vuelos.

La representación digital de video tiene muchos beneficios, incluyendo:

- i. Una arquitectura abierta en los sistemas de video, esto significa la existencia de múltiples resoluciones espacio-temporal, en un mismo flujo de datos.
- ii. Interactividad, permitiendo interrupción para tomar caminos alternativos a través de un video, base de datos, y recuperación de video.
- iii. La demanda en el rango de transmisión.
- iv. La conversión fácil por software de un estándar a otro.
- v. La integración de varias aplicaciones de video, como TV, el videoteléfono, y así sucesivamente, en una plataforma común.
- vi. Editar capacidades, como cortar, pegar, zoom, quitar el ruido.
- vii. Robustez al ruido de canal y la facilidad de encriptación.

Todas estas capacidades requieren procesamiento digital a varios niveles de complejidad.

En el mundo real, todo tiene un costo, y uno de los puntos fuertes de la tecnología digital es el bajo costo. Si el proceso de copiado no genera pérdidas de calidad, las grabadoras digitales no tienen que ser mucho mejores que lo necesario para sobreponerse a ese problema. Sólo necesitan ser adecuadas en la primera generación, cuya calidad será mantenida vez con vez. Ya no hay necesidad, además, de que exista el enorme consumo de cintas de las grandes grabadoras analógicas profesionales; cuando la información consiste solamente de números discretos, éstos pueden ser densamente "empacados" en el medio sin temor a pérdidas de calidad. Si algunos bits cayeran en errores por ruido u omisión, la corrección de errores restauraría los valores originales. Las grabaciones digitales ocupan menos espacio que las analógicas con la misma o mejor calidad.

La manufactura de los circuitos digitales cada vez cuesta menos. La circuitería de dos estados que maneja lenguaje binario puede ser integrada más densamente que la analógica; más funcionalidad en el mismo chip. Los circuitos analógicos son

construidos a partir de un conjunto de componentes que tienen una gran variedad de tamaños y formas, por ende, son caros de ensamblar y de ajustar. La circuitería digital posee diseños de componentes estándar, lo que significa que son más fáciles de ensamblar en equipo automático, y además, se necesita muy poco o nulo ajuste.

1.1.2 Procesamiento de video digital

El procesamiento de video digital se refiere a la manipulación de los datos de video digital. Todas las aplicaciones conocidas de video digital requieren proceso digital para la compresión de datos. Además, algunas aplicaciones pueden beneficiar el proceso adicionando análisis del movimiento, conversión de estándares, perfeccionamiento, y restauración para obtener imágenes de mejor-calidad o extraer un poco de información específica.

El procesamiento digital de imágenes inmóviles ha encontrado uso en el ejército, aplicaciones comerciales, y aplicaciones para nosotros, el consumidor desde los años 60's. Misiones espaciales, imágenes de vigilancia, visión nocturna, tomografía, resonancia magnética, y máquinas de fax son algunos ejemplos. Lo que todavía hace el proceso del video digital diferente a procesar una imagen estática, es que esa imagen de video contiene una cantidad significativa de correlación temporal (redundancia) entre secuencias. Uno puede intentar procesar video como una consecuencia de imágenes, donde cada secuencia se procesa independientemente. Sin embargo, la utilización de la redundancia temporal existente por medio de técnicas de procesamiento de secuencias, nos permite desarrollar algoritmos más eficaces, como el filtrado del movimiento compensado y predicción del movimiento-compensado. Además, algunas tareas, como estimación del movimiento o el análisis de una escena que varía en el tiempo, obviamente no se puede realizar sobre la base de una sola imagen.

A continuación se presentarán las etapas fundamentales para el procesamiento de una imagen:

La primera etapa comprende el proceso de adquisición de imágenes, para esto nosotros necesitamos un sensor de imágenes y la posibilidad de digitalizar la señal producida por el sensor. El sensor puede ser una cámara de barrido de líneas que produzca una línea de la imagen cada vez. Al terminar la adquisición, sigue la etapa de preprocesamiento de la imagen, esto quiere decir, mejorar la imagen por ejemplo mejorar el contraste, eliminar ruido, etc. La siguiente etapa es la de segmentación, la cual se refiere al reconocimiento de caracteres o formas, extrae caracteres individuales y palabras del fondo. A la salida del proceso se tienen los datos del píxel en bruto, que constituyen bien el contorno de una región o bien todos los puntos de una región determinada. La primera decisión que hay que tomar es si los datos se han de representar como un contorno o como una región completa. La elección de una representación completa es sólo una parte de la solución para transformar los datos de píxel en bruto a una forma adecuada para ser posteriormente tratados por computadora. También debe especificarse un método para describir los datos de forma que se resalten los rasgos de interés. A continuación sigue la etapa de la descripción, también denominada selección de rasgos, consiste en extraer rasgos con alguna información cuantitativa de interés o que sean fundamentales para diferenciar una clase de objetos de otra. La última etapa incluye el reconocimiento e interpretación. El reconocimiento es el proceso que asigna una etiqueta a un objeto basándose en la información proporcionada por sus descriptores y la interpretación implica asignar significado a un conjunto de objetos reconocidos, ver figura 1.2.

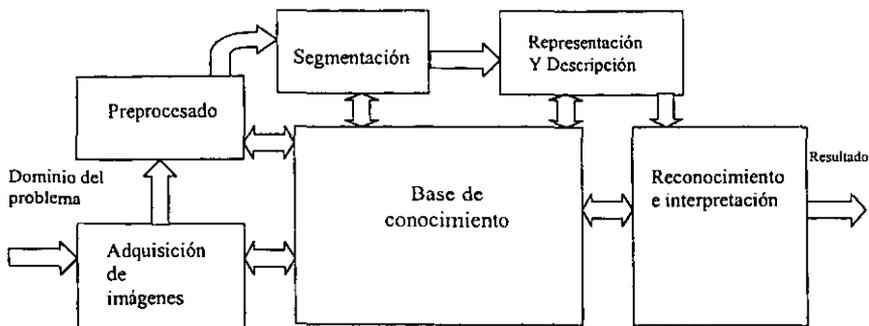


Figura 1.2. Etapas del procesamiento digital de imágenes [3].

1.1.3 Formación geométrica de la imagen

La percepción visual funciona congregando en una escena la luz dispersada por objetos, creando una imagen 2-D (bidimensional). Para poder emplear esta imagen y obtener esta información respectiva acerca de la escena, es necesario comprender el procedimiento geométrico que se da en este proceso.

Cámara de orificio

El procedimiento más simple para producir una imagen es mediante la cámara de orificio (figura 1.3). Sea \mathbf{p} un punto de la escena, con coordenadas (X, Y, Z) y \mathbf{p}' su imagen sobre el plano de imágenes, y con coordenadas (x, y, z) . Si f es la distancia que separa al orificio O del plano de imágenes, con base en la semejanza de triángulos se deducen las siguientes ecuaciones (1.1):

$$\frac{-x}{f} = \frac{X}{Z}, \quad \frac{-y}{f} = \frac{Y}{Z} \Rightarrow x = -\frac{fX}{Z}, \quad y = -\frac{fY}{Z} \quad (1.1)$$

Note que la imagen aparece invertida, tanto de izquierda a derecha como de arriba hacia abajo, en relación con la escena, como lo indica los signos negativos de las ecuaciones. Estas ecuaciones definen un procedimiento de formación de imágenes conocidos como *proyección de perspectiva*.

De la misma manera, también es posible modelar el procedimiento de proyección de perspectiva cuando el plano de proyección se encuentra a una distancia f frente al orificio. En 1435, Alberti recomendó el uso de este procedimiento de formación de imágenes en una superficie de proyección frontal, la que usaban los pintores del renacimiento italiano, como técnica para elaborar representaciones geoméricamente exactas de una escena tridimensional. Para nuestros propósitos la principal ventaja de este modelo es que no produce inversión lateral y elimina los signos negativos de las ecuaciones de perspectiva [4].

En la proyección de perspectiva, las líneas paralelas parecían converger en un punto del horizonte, como si fueran las vías de un tren. Veremos él porqué de lo anterior. De acuerdo con el cálculo vectorial sabemos que un punto cualquiera p_λ de una línea que pase a través de (X_o, Y_o, Z_o) en la dirección que apunta hacia (U, V, W) se define mediante la expresión $(X_o + \lambda U, Y_o + \lambda V, Z_o + \lambda W)$ en la que λ varía entre $+\infty$ y $-\infty$. La proyección de p_λ en el plano de imágenes se expresa de la siguiente manera, ecuación (1.2):

$$\left(f \frac{X_o + \lambda U}{Z_o + \lambda W}, f \frac{Y_o + \lambda V}{Z_o + \lambda W} \right) \quad (1.2)$$

Conforme $\lambda \rightarrow \infty$ o $\lambda \rightarrow -\infty$, lo anterior se aproxima a $p_\infty = (fU/W, fV/W)$ si $W \neq 0$. Denominamos a p_∞ punto de fuga correspondiente a la familia de líneas rectas que están en dirección a (U, V, W) . No importa el punto (X_o, Y_o, Z_o) por el que pase la línea recta en la escena, sólo su dirección.

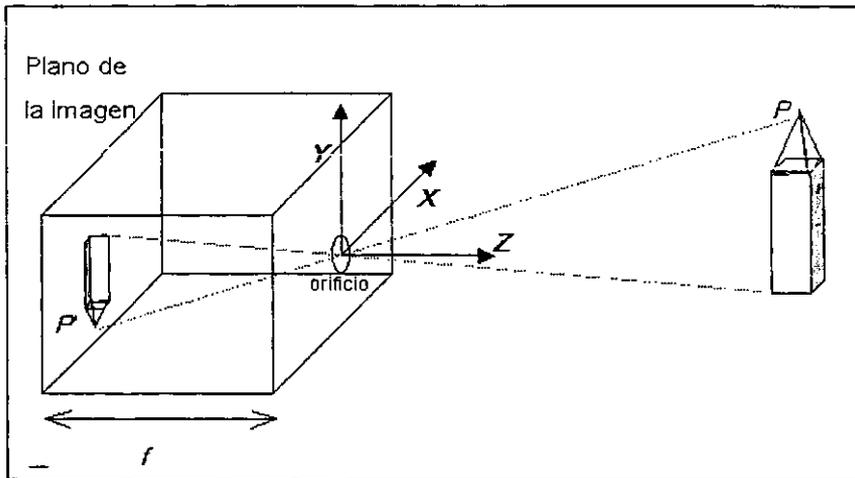


Figura 1.3. Geometría de la formación de imágenes en una cámara de orificio.

Si el objeto tiene poca profundidad en relación con la distancia que lo separa de la cámara, la proyección de perspectiva puede ser aproximada utilizando una proyección ortográfica a escala. La idea es la siguiente: si la profundidad Z de los puntos

del objeto varía un rango $Z_o \pm \Delta Z$, en el que $\Delta Z \ll Z_o$, entonces es posible aproximar el factor de escala f/Z mediante una constante $s=f/Z_o$. Las ecuaciones de la proyección que va de las coordenadas de la escena (X, Y, Z) al plano de la imagen resulta ser $x = sX$ y $y = sY$. Conviene notar que la proyección ortográfica a escala es una aproximación válida sólo para las partes de la escena en donde no hay mucha variación de profundidad interna. No se debe emplear para estudiar "características en su totalidad". Un ejemplo de por qué hay que tener cuidado: en una proyección ortográfica, las líneas paralelas permanecen siendo paralelas en vez de converger en el punto de fuga.

Sistemas de lentes

En los ojos de los vertebrados y en las cámaras fotográficas se utilizan lentes. Estos son más grandes que el orificio de la cámara de la sección anterior, por lo que admiten el paso de mayor cantidad de luz; en compensación, esto impide enfocar con precisión toda la escena. La imagen de un objeto que este a una distancia Z en la escena se produce siempre a la misma distancia del lente Z' , siendo la relación entre Z y Z' la siguiente ecuación:

$$\frac{1}{Z} + \frac{1}{Z'} = \frac{1}{f} \quad (1.3)$$

en donde f es la distancia focal a la lente. Una vez elegida cierta distancia de imagen Z'_o entre el punto nodal de la lente y el plano de imagen, los puntos de la escena cuyas profundidades sean de un valor alrededor de Z_o , en donde Z_o es la distancia al objeto, se producirá una imagen razonablemente bien enfocada. A este rango de profundidades de la escena se le conoce como *profundidad de campo* [4].

- Conviene notar que debido a que es característico que la distancia al objeto Z sea mucho mayor que la distancia a la imagen Z' o f , generalmente hay que efectuar la siguiente aproximación:

$$\frac{1}{Z} + \frac{1}{Z'} \approx \frac{1}{Z'} \Rightarrow \frac{1}{Z'} \approx \frac{1}{f} \quad (1.4)$$

De esta manera, la distancia de imagen es $Z' \approx f$. Por lo tanto, podemos seguir utilizando las ecuaciones de la proyección en perspectiva de la cámara de orificio para explicar el procedimiento geométrico de formación de imágenes en un sistema de lentes.

Para poder enfocar objetos a diferentes distancias Z , el lente del ojo (véase figura 1.4) tiene que modificar su forma, en tanto que la lente de una cámara se desplaza a lo largo de la dirección Z .

El plano de la imagen está recubierto de un material fotosensible constituido por:

Haluro de plata, en el caso de las películas fotográficas.

Rodopsina y sus variantes, en el caso de la retina.

Circuitos de silicio en la cámara CCD (sensor acoplado por carga), artefacto de acoplamiento de carga. En cada sitio de una CCD se integran durante cierto lapsos los electrones liberados por absorción fotónica.

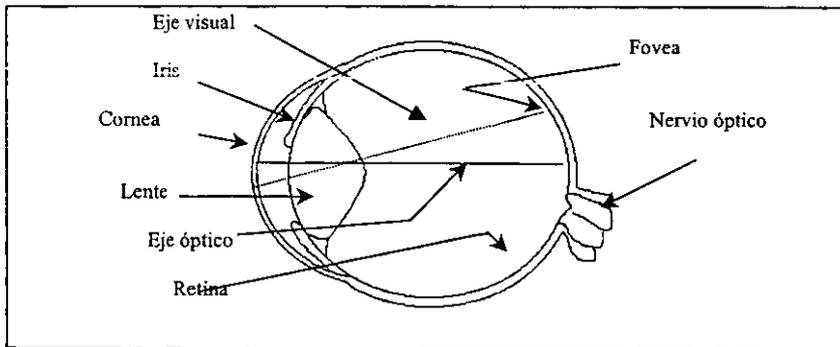


Figura 1.4. Sección transversal horizontal del ojo humano.

En el ojo y en la cámara CCD, el plano de imagen se subdivide en píxeles: por lo general 512×512 (0.25×10^6) en la cámara CCD, dispuestos sobre una cuadrícula rectangular; 120×10^6 bastoncillos y 6×10^6 conos, en el ojo, dispuestos en un mosaico hexagonal.

En ambos casos, la señal detectada en el plano de imágenes se modela de acuerdo con la variación de la brillantez de la imagen a través del tiempo: $I(x, y, t)$.

1.1.4 Formación fotométrica de la imagen

La brillantez de píxel p en la imagen es proporcional a la cantidad de luz dirigida hacia la cámara por la parte de la superficie S_p proyectada hacia el píxel p . Lo anterior, a su vez, dependerá de las características de reflectancia de S_p , de la posición y distribución de las fuentes luminosas. También depende en parte de las características de reflectancia del resto de la escena, ya que existe la posibilidad de que otras superficies de la escena opere como fuentes de luz indirecta al reflejar la luz que les envía S_p .

La luz reflejada desde un objeto puede ser caracterizada como de tipo difuso o de tipo especular. En primer caso, la luz penetra bajo la superficie de un objeto, la absorbe y vuelve a emitirla. La superficie ofrecerá la misma brillantez a un observador, desde cualquier dirección que mire. Para describir la reflexión de la luz desde una superficie perfectamente difusora o *lambertiana* se utiliza la ley del coseno de Lambert [4]. La intensidad, E , de la luz reflejada desde un difusor perfecto está dada por la siguiente ecuación (1.5):

$$E = \rho E_o \cos \theta \quad (1.5)$$

En donde E_o es la intensidad de la fuente luminosa; ρ es el albedo, cuyo valor va de 0 (superficies perfectamente negras) a 1 (superficies perfectamente blancas); y θ es ángulo formado por la dirección de la luz y la superficie normal.

La luz reflejada de manera especular procede de la superficie externa de un objeto. En este caso la energía de la luz reflejada se concentra básicamente en una determinada dirección; aquélla donde el rayo reflejado se encuentra en el mismo plano del rayo incidente y se cumple la condición de que el ángulo de reflexión es igual al ángulo de incidencia. Es así como funciona un espejo perfecto.

En la vida real, las superficies se comportan combinando características de reflexión difusa y especular. El modelado de lo anterior en una computadora es el pan cotidiano de las gráficas por computadora. La obtención de imágenes realistas es lograda generalmente mediante el rastreo de rayos, en el que se trata de simular el procedimiento físico de la luz originada en fuentes luminosas y que se refleja y vuelve a reflejarse varias veces. El problema de la percepción por computadora que consiste en definir una forma a partir del sombreado es el de invertir la situación anterior, es decir, a partir de la imagen "producida" y tratando de deducir la distribución de la escena tridimensional que la produjo.

Espectrofotometría de la formación de imágenes

Hemos hablado de la intensidad de las imágenes, $I(x, y, t)$, ignorando por completo el hecho de que la luz visible está formada por toda la gama de longitudes de onda que van de los 400nm en el extremo violeta del espectro hasta los 700nm en extremo rojo. Considerando que existe un rango continuo de longitudes de onda, ¿por qué se dice que contamos con tres colores primarios? La razón es que el color es algo que existe inherentemente en el ojo de cada quien. En el ojo existen tres tipos diferentes de conos, con tres curvas de sensibilidad espectral distintas $R_k(\lambda)$. La salida del cono número k en la ubicación (x, y) en el momento t es, por tanto, $I_k(x, y, t) = \int I(x, y, t, \lambda) R_k(\lambda) d\lambda$. El espacio infinito constituido por las longitudes de onda se proyecta en un espacio de color tridimensional. Lo anterior quiere decir que debemos considerar a I como un vector tridimensional que está en (x, y, t) . Dado que el ojo correlaciona distintos espectros de frecuencia con la misma percepción de color, es natural la existencia de *metámeros*; distintos espectros luminosos que al ojo humano parecen iguales.

1.2 Flujo óptico

Si la cámara se desplaza en relación con una escena tridimensional, el cambio aparente producido en la imagen se conoce como flujo óptico. Lo anterior describe la dirección y la

velocidad de movimiento de las características de la imagen como consecuencia del movimiento relativo que se da entre el observador y la escena. El flujo óptico codifica información útil sobre la estructura de la escena. Por ejemplo cuando se observan desde un carro que se desplaza, los objetos distantes aparentemente se desplazan mucho más lentamente que los objetos que están más cerca; por lo tanto, la velocidad del movimiento aparente puede darnos cierta información sobre la distancia.

1.2.1 Movimiento 2D vs. movimiento aparente

La estimación de movimiento se refiere a estimar el movimiento de un objeto 3D en una imagen plana 2D, éste es uno de los problemas fundamentales de procesamiento digital de imágenes. Para representar las imágenes que varían en el tiempo, las escenas en 3D son proyecciones al plano 2D. Lo que se desea es estimar el movimiento en el plano 2D (al mismo tiempo velocidad y desplazamiento) imágenes que varían con el tiempo. Sin embargo, la velocidad en el plano 2D no siempre es observable por muchas razones que mencionaremos en este capítulo. En cambio, lo que nosotros observamos es lo que llamamos movimiento aparente (flujo óptico o correspondencia). Este punto es importante ponerlo en claro; la velocidad 2D, el flujo óptico, desplazamiento en el plano 2D y sus campos correspondientes.

Movimiento 2D

El movimiento en 2D es también llamado “movimiento proyectado”, esto se refiere a la perspectiva o la proyección del movimiento 3D en una imagen plana. El movimiento 3D puede ser caracterizado en cualquiera de los dos términos de velocidad instantánea en el plano 3D o un objeto de desplazamiento 3D. Las expresiones para las proyecciones del desplazamiento 3D y los vectores de velocidad asumen un movimiento rígido, en la imagen plana.

El concepto de vector de desplazamiento 2D es ilustrado en la figura 1.5. El objeto en el punto P en el tiempo t se mueve hacia P' en el tiempo t' . La proyección de el desplazamiento del punto P y P' en la imagen dan los puntos respectivos p y p' . La figura 1.5, describe una vista 2D del movimiento en el punto p en el tiempo t hacia p' en el tiempo t' como la proyección correspondiente del movimiento 3D. Notamos que si estos puntos se encuentran en la línea dará el mismo desplazamiento del vector 2D.

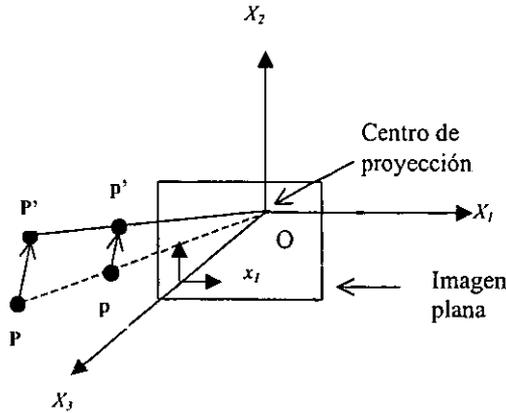


Figura 1.5. Movimiento en 3D contra 2D.

El desplazamiento proyectado entre los tiempos t y $t' = t + \ell \Delta t$ donde ℓ es un entero y Δt es un intervalo temporal, puede ser definido para todo $(x, t) \in \mathbb{R}^3$, resultando un valor real de desplazamiento 2D, el vector sería $d_c(x, t; \ell \Delta t)$ en el espacio temporal. El vector de desplazamiento 2D se refiere a la simple representación de esta función dada por

$$d_p(x, t; \ell \Delta t) = d_c(x, t; \ell \Delta t) \quad (\text{T.6})$$

Correspondencia y flujo óptico

El desplazamiento de una imagen plana en la coordenada X de un tiempo t a t' , basadas en la variaciones de $s_c(x, t)$, es

llamada un vector de correspondencia. Un vector de flujo óptico es definido como un rango temporal de cambios de coordenadas de una imagen plana, $(v_1, v_2) = (dx_1/dt, dx_2/dt)$ en un punto particular $(x, t) \in \mathbf{R}^3$ determinado en el espacio temporal del modelo de intensidad $s_c(x, t)$ [2]. Esto corresponde a un vector de velocidad de un píxel instantáneo. (Teóricamente, el flujo óptico y vector de correspondencia son idénticos en el límite $\Delta t = t' - t$ que tiende a 0). En la práctica, nosotros definimos la correspondencia (flujo óptico) como un vector de desplazamientos de píxeles (velocidad) basada en las variaciones observables de la imagen 2D en un espacio temporal. El campo de correspondencia y el flujo óptico son conocidos como el “desplazamiento aparente 2D”.

La correspondencia (flujo óptico) es en general, diferente del desplazamiento 2D (velocidad 2D) debido a:

Falta de suficiente gradiente espacial en la imagen: Esto quiere decir que debe haber una suficiente variación de niveles de gris (color), si la región no se está moviendo por el actual movimiento, como se muestra en la figura 1.6, donde un círculo con intensidad uniforme rota a través de su centro. Este movimiento genera un flujo no óptico, y esto no se puede observar.

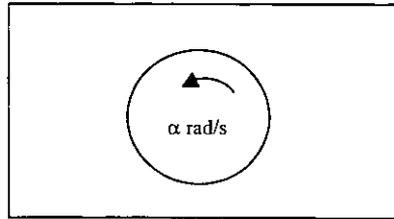


Figura 1.6. Todo el movimiento proyectado no genera flujo óptico.

Cambios en una iluminación externa: Un flujo óptico observable puede no ser siempre correspondiente a un movimiento actual. Por ejemplo, si la iluminación externa varía de una secuencia a otra secuencia, como se muestra en la figura 1.7, entonces un flujo óptico será observado aunque no haya movimiento. Por lo tanto hay cambios en la iluminación externa impar del actual campo de movimiento 2D.

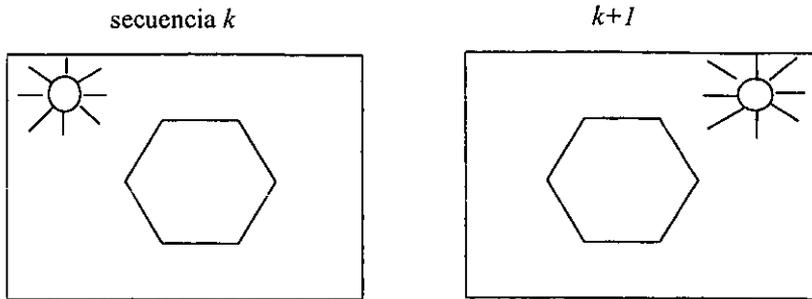


Figura 1.7. Todo flujo óptico no corresponde a un movimiento proyectado.

En algunos casos, el sombreado puede variar de una secuencia a otra secuencia aunque no haya un cambio en la iluminación externa. Por ejemplo, si un objeto rota su cambio de superficie normal, el resultado es un cambio en el sombreado. Este cambio en el sombreado puede causar cambio en la intensidad en los píxeles entre un movimiento de trayectoria, el cual necesita ser tomado en cuenta en la estimación en movimiento.

En conclusión, el desplazamiento 2D y los campos de velocidad son proyecciones de los campos respectivos 3D en una imagen plana, mientras que la correspondencia y el flujo óptico son funciones de velocidad y desplazamiento percibidas desde la variación del tiempo acompañada de una imagen de intensidad.

1.2.2 Estimación 2D del movimiento

El problema de estimación de movimiento 2D puede ser planteado como:

- 1) La estimación de los vectores de correspondencia de una imagen plana $\mathbf{d}(\mathbf{x}, t; \ell \Delta t) = [d_1(\mathbf{x}, t, \ell \Delta t) \ d_2(\mathbf{x}, t, \ell \Delta t)]^T$ entre los tiempos t y $t + \ell \Delta t$, para todo $(\mathbf{x}, t) \in \wedge^3$ y ℓ es un entero, o
- 2) La estimación de los vectores del flujo óptico $\mathbf{v}(\mathbf{x}, t) = [v_1(\mathbf{x}, t) \ v_2(\mathbf{x}, t)]^T$ para todo $(\mathbf{x}, t) \in \wedge^3$.

Los vectores de correspondencia y flujo óptico usualmente varían de un píxel a otro píxel (variación de movimiento en el espacio) debido a la rotación de objetos en la escena, y a una función de tiempo debido a la aceleración de objetos.

El problema de la correspondencia: El problema de la correspondencia puede ser establecido como un problema de estimación de movimiento forward o backward (adelante o atrás), dependiendo en tiempo del vector de movimiento, el cuál es definido desde el tiempo t a $t+\ell\Delta t$ o desde t a $t-\ell\Delta t$, como se muestra en la figura 1.8.

Estimación Forward: Dadas las muestras espacio temporal $s_p(\mathbf{x}, t)$ en el tiempo t y $t+\ell\Delta t$, que están relacionadas por

$$s_p(x_1, x_2, t) = S_c(x_1 + d_1(\mathbf{x}, t; \ell\Delta t), x_2 + d_2(\mathbf{x}, t; \ell\Delta t), t + \ell\Delta t) \quad (1.7)$$

O su equivalente,

$$s_k(x_1, x_2) = s_{k+\ell}(x_1 + d_1(\mathbf{x}), x_2 + d_2(\mathbf{x})) \text{ tal que } t = k\Delta t \quad (1.8)$$

Encontrar el valor real del vector de correspondencia $\mathbf{d}(\mathbf{x}) = [d_1(\mathbf{x}) \ d_2(\mathbf{x})]^T$, donde los argumentos temporal de $\mathbf{d}(\mathbf{x})$ son establecidos.

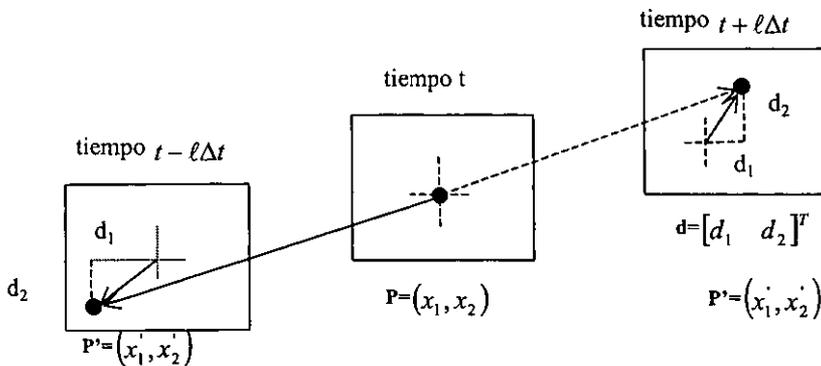


Figura 1.8. Estimación correspondiente en Forward y Backward.

Estimación Backward: Si nosotros definimos los vectores de correspondencia desde el tiempo t a $t + \ell\Delta t$ entonces, el modelo de movimiento 2D se vuelve

$$s_k(x_1, x_2) = s_{k-\ell}(x_1 + d_1(x), x_2 + d_2(x)), \quad \text{tal que } t = k\Delta t \quad (1.9)$$

Alternativamente, el vector de movimiento puede ser definido desde el tiempo $t + \ell\Delta t$ a t . Entonces nosotros tenemos.

$$s_k(x_1, x_2) = s_{k-\ell}(x_1 - d_1(x), x_2 - d_2(x)), \quad \text{tal que } t = k\Delta t \quad (1.10)$$

Aunque discutimos ambos tipos de estimación de movimiento, la estimación de movimiento backward es más conveniente para la compensación de movimiento y forward es cómodamente empleada en la comprensión de video predictivo. Observe que debido a que $x \pm d(x)$ generalmente no corresponde a un lugar en el dominio de las imágenes, el lado derecho de las expresiones está dado en términos del video continuo, que no es conocido ($d(x)$ puede ser real). Ahora más métodos de estimación de correspondencia incorporan un esquema de interpolación [Anexo A]. El problema de correspondencia también se origina en la estimación de la disparidad estéreo, donde nosotros tenemos un par de imágenes izquierdo-derecho en lugar de un par de imágenes temporales.

Registro de imagen: El problema del registro es un caso especial de problema de correspondencia, en donde las dos secuencias son globalmente desplazadas con respecto la una de la otra, por ejemplo, múltiples exposiciones de una escena estática con una cámara trasladada.

Estimación de flujo óptico: Dadas las muestra $s_p(x_1, x_2, t)$ en un espacio 3D \wedge^3 , determinar la velocidad 2D $v(x, t)$ para todo $(x, t) \in \wedge^3$. Es claro que la estimación del flujo óptico y la correspondencia de vectores desde dos secuencias son equivalentes, con $d(x, t; \ell\Delta t) = v(x, t)\ell\Delta t$, asumiendo que la velocidad es constante durante un intervalo de tiempo $\ell\Delta t$. Note que en la presencia de aceleración necesitamos considerar más de dos secuencias para estimar el flujo óptico en un tiempo.

La estimación de movimiento 2D, expresado ya sea como la correspondencia o la estimación del flujo óptico, basado sólo en dos secuencias, es un problema “mal propuesto” en la ausencia de suposiciones adicionales acerca de la naturaleza del movimiento. Se tiene un problema “mal propuesto” cuando no existe una solución única. La estimación de movimiento 2D sufre problemas de existencia, solución única y continuidad.

- Existencia de una solución: No se puede establecer correspondencia, entre puntos cubiertos y no cubiertos del fondo. Este es conocido como el problema de *oclusión*.
- Solución única: Si las componentes del desplazamiento (o velocidad) en cada píxel son tratadas como variables independientes, entonces el número de incógnitas es dos veces el número de observaciones (los elementos de la diferencia de la secuencia). Esto se llama el problema de *apertura*.
- Continuidad de la solución. La estimación de movimiento es muy sensible a la presencia del ruido en la observación en imágenes de video. Una pequeña cantidad de ruido resulta en una larga desviación en el movimiento estimado.

Problema de oclusión

La oclusión se refiere a cubrir o no cubrir una superficie debido a la rotación y translación 3D de un objeto el cual ocupa sólo una parte del campo de visión. Los conceptos de cubrir y no cubrir el fondo son ilustrados en la figura 1.9. donde el objeto, indicado por las líneas sólidas, se traslada en la dirección x_1 desde el tiempo t a t' . El índice de la secuencia del tiempo t y t' es k y $k+1$, respectivamente. La línea de puntos en la secuencia k indica el fondo para ser cubierto en una secuencia $k+1$. Entonces, no es posible encontrar una correspondencia para estos píxeles en la secuencia $k+1$. La línea de puntos de la región de la secuencia $k+1$ indica que no se cubrió el fondo por el movimiento del objeto. No hay correspondencia para estos píxeles en la secuencia k .

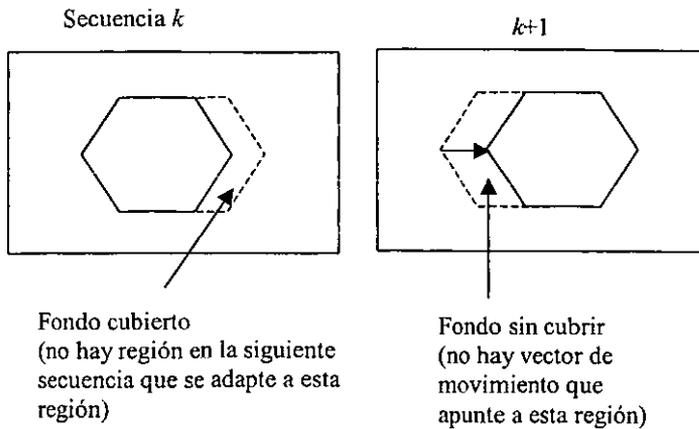


Figura 1.9. Problema del fondo cubierta y no cubierta.

Modelos de campos de movimiento en 2D

Los algoritmos de estimación de movimiento necesitan asumir (modelos) acerca de la estructura del campo de movimiento 2D. A continuación se provee una breve descripción de estos modelos:

Modelos paramétricos

Los modelos paramétricos describen la proyección ortográfica o la perspectiva del movimiento 3D (desplazamiento y velocidad) de una superficie en una imagen plana. En general, el movimiento paramétrico 2D depende de una representación de la superficie 3D. Por ejemplo, un campo de movimiento desde el movimiento rígido 3D de una superficie plana bajo proyección ortográfica puede ser representada por 6 parámetros afines al modelo, mientras que bajo una proyección de perspectiva puede ser representada por 8 parámetros de un modelo no lineal [2].

Una subclase de modelos paramétricos es llamada modelos cuasi-paramétricos, la cual trata la profundidad de cada punto independiente 3D. Entonces los 6 parámetros del movimiento 3D obligan al vector de flujo de la imagen local a quedarse entre una

línea específica, mientras sabemos el valor de la profundidad local la cual es requerida para determinar el valor exacto del vector de movimiento. Estos modelos pueden servir para obligar a regular la estimación de los vectores de movimiento 2D, que permiten formular estimaciones de movimiento simultáneas 2D y 3D.

Modelos no paramétricos

La principal desventaja de los modelos paramétricos es que solo es aplicable en el caso de movimiento rígido 3D. Lo siguiente es una descripción breve.

- **Métodos basados en la ecuación de flujo óptico (OFE):** Estos métodos estiman el flujo óptico en términos de intensidad del gradiente en una imagen espacio-temporal. Con imágenes monocromáticas, el OFE necesita ser usado en conjunción con una restricción de suavizado espacio-temporal apropiada, la cual requiere que el desplazamiento del vector varíe lentamente sobre el fondo. Con imágenes de colores el OFE puede ser impuesta en cada banda de color separadamente, la cual podría posiblemente obligar al vector de desplazamiento tener tres diferentes direcciones.
- **Modelo de movimiento por bloque.** Este asume que la imagen está compuesta por bloques que se mueven. Se utilizan dos métodos para determinar el desplazamiento de bloques de una secuencia a otra secuencia: los métodos de correlación de fase y block matching. En el método de correlación de fase, el término lineal de la diferencia de fase de Fourier entre dos secuencias consecutivas determina el movimiento estimado. Block matching busca la localización del mejor lugar de emparejamiento de bloques de un tamaño fijo en la secuencia siguiente o previa basada en el criterio de la distancia. La forma básica de ambos métodos se aplica solo en el movimiento trasladado.
- **Método Pel-recursivo:** El método Pel-recursivo es un predictor-correctivo de estimación de desplazamiento. La predicción puede ser tomada como un valor del movimiento estimado en una localización previa del píxel o como una combinación de movimiento estimado en una región vecina de un píxel.

- **Métodos Bayesianos:** El método bayesiano utiliza restricciones probabilísticas de suavizado, usualmente en la forma del campo aleatorio de Gibbs para estimar el campo de desplazamiento. Su principal desventaja es lo extensivo de la computación que se requiere.

1.3 Sistemas de visión estéreo

¿Por qué necesitamos nosotros video 3D? La respuesta inmediata es que video 3D es la "Alta fidelidad" en video porque los métodos de visión estéreo pueden agregar percepción de 3D hacia 2D.



Figura 1.10. Sistema que integra la percepción del oído.

Recientemente, las tecnologías 2D de comunicación visuales han madurado tan rápido que varios sistemas comerciales están disponibles, la comunicación visual está basada en normas como JPEG, MPEG I y II, o H.263. No obstante, estas normas no podrían estar suficientemente acondicionadas a las nuevas demandas crecientes para captar el realismo o las representaciones más naturales de la escena.

Para reunir estos requisitos los nuevos sistemas tienen que integrar varios de los sentidos humanos como la vista, oído (ver figura 1.10), tacto, olfato, y gusto. La imaginación que las tecnologías han traído ha permitido desarrollar esas capacidades. Por ejemplo, la primera fase de televisión de B/W fue diseñada para extender el sentido de la vista y oído y se integraron colores para representar escenas realistas. Las recientes actividades en HDTV son otro esfuerzo para agregar realismo en el despliegue 2D. Los Nuevos desarrollos de la visión estéreo pueden abrir una nueva forma de desplegar imágenes y del video en 3D. Así, la televisión en 3D, sin los lentes especiales, será el próximo paso del sistema de la televisión. En el futuro, varias tecnologías como televisión 3D, teléfono y computación se esperan que ofrezcan la

funcionalidad interactiva combinado con lo que proporciona la "realidad virtual"

En estéreo (o multi-vista) se ligan imágenes para tener un mundo de aplicaciones principalmente en los campos de visualización 3D (CAD), telemedicina en 3D, telerrobótica 3D, comunicaciones visuales 3D, televisión 3D, cine, y la Realidad Virtual.

Hay varias maneras de reconocer escenas 3D; la Holografía, monitor 2D con lentes, despliegue "montados sobre la cabeza", y despliegues estéreo 3D. La idea básica de ver cuadros en estéreo es que eso permite a cada ojo ver la imagen correspondiente, respectivamente.

- Vista libre: percibe 3D sin usar dispositivos especiales como estereoscopios, los lentes "Anaglyph" (rojo/verde), lentes polarizados, o equipo de VR. Usando el "ojo al desnudo", ver las imágenes en paralelo o vista cruzada para producir el efecto 3D, ver figura 1.11.



Figura 1.11. Métodos vista libre.

- Anaglyph: Es una imagen estéreo que requiere lentes con rojo y verde (o azul) para 3D . Las dos imágenes estéreo están impresas encima de cada una, pero con un desplazamiento. Tradicionalmente, la imagen para el ojo izquierdo está impresa en tinta roja y la imagen del ojo correcta está impresa en tinta verde.
- Monitor con lentes especiales: para percibir la escena 3D en un monitor 2D, nosotros tenemos que producir o sintetizar imágenes 2D por medio de multiplexación temporal o método de la polarización. Se sincronizan lentes especiales equipados con contraventanas de cristal de líquido que cambian de opaco a claro electrónicamente. Como resultado, cada ojo sólo ve respectivamente la imagen de las cámaras correspondientes. Para un monitor con un refresco de 60 Hz, en un sistema

basado en NTSC, cada ojo ve la imagen a un refresco proporcional de 30 Hz.

Despliegue estéreo

1. Monitor de Barrera paralela: El cuadro detrás de la pantalla de la barrera paralela está compuesto de imagen 2D de diferentes ángulos. Cada ojo ve la imagen correspondiente a través de las rayas apropiadas percibiendo 3D.
2. Monitor lenticular: las lentes usan plástico delgado, puesto encima de la fotografía restringe la vista de cada ojo a una parte particular del cuadro.
3. Problemas: puede haber efectos colaterales como manchas en el ojo y dolores de cabeza en algunos usuarios. Además, el campo angular de vista está limitado. Los sistemas de televisión 3D no tienen ninguna oportunidad de ganar popularidad a la televisión, común, si se introducen en el mercado sin resolver estos inconvenientes.

Se ha demostrado que un sistema de despliegue que le exige al espectador que lleve lentes especiales, nunca tendrá éxito en el mercado de entretenimiento casero. La visión estéreo puede ser una solución. El prototipo de un despliegue en visión estéreo, se basó en técnicas de la pantalla lenticulares, que ha sido desarrollado por el NHK (Japón) y Universidad de Cambridge (Reino Unido).

1.3.1 Aplicaciones de la Estereoscopia

Actualmente diversos campos científicos y técnicos se benefician de la estereoscopia. Por citar algunos:

Topografía y estudio del terreno

Una de las aplicaciones prácticas más antigua es la visualización y medición del relieve terrestre mediante fotografías aéreas. Si un avión toma dos fotografías de una zona de un terreno con una cierta distancia calculada entre ellas, se obtiene un par

estéreo, que posteriormente puede verse en relieve con un estereoscopio especial. Si las tomas se realizan con la adecuada precisión, permiten calcular elevaciones en el terreno, para lo cual se emplean los estéreo-comparadores. En la actualidad, en fotogrametría, esta labor se puede realizar con gran precisión y con visión estéreo gracias a estaciones y software especialmente diseñados, como los de Intergraph y Zeiss. A partir de datos del terreno, pueden también generarse imágenes 3D simuladas mediante software específico, por ejemplo para representar el relieve submarino. Otro ejemplo es el trabajo topográfico realizado en Febrero del 2000 desde el trasbordador espacial Endeavour, dentro del proyecto SRTM, que permite obtener mapas tridimensionales de una resolución extraordinaria.

Estudio de la tierra y otros planetas

De forma similar a la fotografía aérea, NASA ha obtenido numerosas vistas tridimensionales de fotografías de la Tierra obtenidas desde satélites [5], así como también de otros planetas de nuestro Sistema Solar. Las extraordinarias imágenes estéreo de la superficie de Marte obtenidas por la sonda Pathfinder de la NASA son otro ejemplo de aplicaciones para el estudio de otros planetas. La toma de imágenes en estéreo no solo sirvió para ver la superficie de Marte en 3D, sino para calcular distancias y tamaños de las rocas y conducir con más seguridad el vehículo [6].

CAD (Diseño Asistido por Computadora) y CAE (Ingeniería Asistida por Computadora)

Es una poderosa herramienta para diseño y visualización de prototipos, por ejemplo en la industria automovilística. Chrysler, Ford, Opel, Renault, Volvo y otros fabricantes ya usan estas técnicas, con un importante ahorro en tiempo y dinero durante el desarrollo [7]. Los más importantes paquetes y estaciones de diseño por ordenador, como IBM, HP, DEC, Sun o Silicon Graphics, soportan actualmente la visualización estereoscópica mediante gafas LCS, como las de Stereographics o VRex.

Medicina

Es uno de los campos en los que la estereoscopia proporciona más ayuda para la enseñanza, la interpretación de imágenes para el diagnóstico o como ayuda en las intervenciones.

No es una novedad que lupas y microscopios de precisión cuentan con visión estéreo desde hace tiempo. Firmas como Zeiss u Olympus disponen de diversos modelos según las aplicaciones. Si a un microscopio estéreo se le conectan dos cámaras de vídeo, se puede ofrecer una presentación 3D en un monitor o pantalla grande de vídeo, así como grabar las imágenes 3D.

El campo de la microcirugía ofrece grandes posibilidades. Zeiss cuenta ya con sistemas de microcirugía tridimensional, como el MediLive 3D [8], del que ya existen referencias sobre sus ventajas aplicado a la oftalmología. También VRex cuenta con un sistema de microcirugía orientado a la endodoncia. Estos sistemas usan un multiplexor para entrelazar las imágenes izquierda y derecha, y la visualización tridimensional se consigue con gafas de cristal líquido (LCS).

También la endoscopia cuenta con una gran ayuda basada en la estereoscopia. El sistema Endolive de Zeiss es un ejemplo de ello.

Una ventaja de este sistema es que todo el equipo quirúrgico puede observar en una gran pantalla y en 3D una intervención si está dotado de gafas para la visión estereoscópica. Además las imágenes tridimensionales pueden grabarse en un vídeo convencional para estudiarlas posteriormente o emplearlas en la docencia.

En la enseñanza tiene evidentes aplicaciones en la visualización de muestras y en la creación de programas multimedia de anatomía virtual. En la Escuela de Medicina de Los Angeles, Universidad de California hay algunos ejemplos [9].

También se usa para visualizar imágenes o modelos del interior del cuerpo humano, bien artificiales, bien generados a partir de imágenes reales obtenidas por medio de TAC (Tomografía Asistida por Computador) o RMN (Resonancia Magnética Nuclear). Técnicas como la radiografía estereoscópica

permiten situar claramente cuerpos extraños o anomalías en el interior del paciente.

Asimismo se le encontrado aplicaciones para el diagnóstico de enfermedades oculares, (un ejemplo en Kasha Software, Inc.), o para distraer a los pacientes en odontología, utilizando un HMD (Head Mounted Display), un sistema diseñado inicialmente para realidad virtual.

Ingeniería Molecular

En ingeniería molecular, sin la visualización estéreo en estaciones de diseño sería muy difícil crear nuevas moléculas complejas. Puede ver moléculas en estéreo en el Departamento de Química de la Widener University [10].

Telepresencia

Sistemas de video-cámaras estéreo permiten operar en entornos peligrosos u hostiles con la máxima precisión. La NASA ya tiene cierta experiencia respecto a sistemas de telepresencia submarina [11].

Realidad Virtual

La técnica denominada Realidad Virtual básicamente es una interacción usuario-ordenador en la que se generan las imágenes estereoscópicas en tiempo real, introduciendo al espectador en un escenario 3D artificial. Por citar algunas, se encuentran las siguientes aplicaciones:

En *arquitectura*, donde la Realidad Virtual nos permite navegar por el interior de un edificio antes de que se construya.

En *arqueología*, permite recrear edificios y ciudades de viejas civilizaciones, o ayudar en la restauración de monumentos.

En *medicina*, es posible simular intervenciones quirúrgicas o navegar por el interior del cuerpo humano para planificar operaciones o en la enseñanza. En la Escuela de Medicina de San Diego de la Universidad de California ya se experimenta con ello

[12]. Los sistemas llamados "Realidad Aumentada" superponen a una imagen real otra generada por ordenador. Esto permite que el cirujano vea sobre la zona de intervención una imagen sintética tridimensional, con indicaciones precisas en un punto de interés especial. Puede ser de gran ayuda en operaciones delicadas, como por ejemplo en el cerebro.

En la *industria automovilística*, es posible situarse al volante de un automóvil antes de fabricarlo.

En la *industria aeroespacial*, en simuladores de vuelo de aviones o para simular entornos de naves espaciales u operaciones en el espacio.

La Realidad Virtual es un tema que excede por sí solo el propósito y la capacidad de este trabajo.

Manufactura

Webber Metals Corporation, es una empresa de ensamblaje, que incremento sus ventas debido a las tendencias de modernización de su laboratorio para su examen de control de calidad. El fabricante cambió del examen mecánico convencional (CMM) a una máquina sin contacto con visión. Como respuesta, levantaron el rendimiento de procesamiento 10:1. Estos exámenes tienen la ventaja de muestreo, además la capacidad de realizar el examen sin contacto con visión en las piezas, que generan automáticamente el informe de ejecución del examen y alcanzan una calidad más alta de la parte total.

CAPÍTULO 2

Estimación de movimiento basada en el gradiente

En este capítulo abordaremos los conocimientos suficientes para modelar el movimiento de un objeto en 3D y proyectarlo al plano 2D. El movimiento 2D describe la relación entre regiones de imágenes sucesivas y proporciona un conjunto reducido de parámetros. Aunado a todo esto, se integra un algoritmo que manipula regiones de las imágenes de análisis que permite reducir el tiempo de transmisión de los resultados e integrándolo a un sistema de codificación a bajas tasas de bit, nos permite un rápido acceso a los datos. Esto es importante ya que los servicios actuales requieren compresión de vídeo con bajas tasas de bit ya que se utilizan en gran medida en las líneas de transmisión en banda base con anchos de banda estrechos. Por ejemplo, internet, videotelefono y servicios de comunicaciones móviles que llegan al público mediante el uso tradicional de la red pública telefónica conmutada. Por eso la codificación digital de secuencias de video con bajas tasas de bit sigue teniendo un gran campo de trabajo. Este tema por su extensión sale del objetivo de este trabajo. Para lograr tasas de bit bajas se necesita un conjunto reducido de parámetros que representen una gran cantidad de información.

Estos modelos son herramientas que van a permitir estimar campos de vectores de desplazamiento (o de velocidades) conocido también como flujo óptico, útil para los diferentes tipos

de aplicaciones como la codificación por compensación de movimiento, el análisis del movimiento y el emparejamiento estéreo. Hacemos referencia a diferentes modelos de movimiento, pero nos enfocaremos al modelo lineal el cual tiene cuatro parámetros tx , ty , k y θ . La importancia en este trabajo es el estudio e implementación del algoritmo del gradiente para la estimación de movimiento basada en regiones utilizando un estimador cuadrático.

2.1 Modelo de movimiento 3D

Repaso de geometría:

El desplazamiento en 3D de un objeto rígido entre los instantes t_1 y t_2 puede descomponerse (utilizando un movimiento del plano general) en seis movimientos elementales: tres traslaciones y tres rotaciones.

Si uno sólo conoce la posición de un objeto en dos instantes t_1 y t_2 , no es posible determinar la trayectoria real del objeto porque existe una infinidad de diferentes posibles trayectorias que cubren estas condiciones. Relativamente, los seis grados de libertad corresponden a tres valores únicos para los ángulos de rotación suponiendo centros fijos de rotación (es decir se escoge el centro de gravedad de cada región) y tres coeficientes de translación. El resultado de esto es que siempre es posible analizar cualquier desplazamiento de un objeto rígido como la suma de los desplazamientos correspondientes a los seis movimientos elementales usando un movimiento plano general [13].

Uno puede definir el desplazamiento 3D de cada punto como $P(X_i, Y_i, Z_i)$ en el instante i , de un objeto rígido animado de cualquier movimiento de translación y de una rotación alrededor del centro de gravedad del objeto entre los momentos t_1 y t_2 de la manera siguiente:

Si uno nota que θ_x (respectivamente θ_y , θ_z), son los ángulos de rotación elementales alrededor del eje x (respectivamente y , z), T_x , T_y , T_z las coordenadas del vector de translación y X_{gi} , Y_{gi} , Z_{gi} ,

las coordenadas del centro de gravedad G_i del plano 3D en el instante i , entonces tenemos:

$$\begin{pmatrix} X_2 - X_{g1} \\ Y_2 - Y_{g1} \\ Z_2 - Z_{g1} \end{pmatrix} = \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} + R_x \cdot R_y \cdot R_z \begin{pmatrix} X_1 - X_{g1} \\ Y_1 - Y_{g1} \\ Z_1 - Z_{g1} \end{pmatrix} \quad (2.1)$$

con

$$R_z R_y R_x = \begin{pmatrix} \cos\theta_z & -\text{sen}\theta_z & 0 \\ \text{sen}\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_y & 0 & -\text{sen}\theta_y \\ 0 & 1 & 0 \\ \text{sen}\theta_y & 0 & \cos\theta_y \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\text{sen}\theta_x \\ 0 & \text{sen}\theta_x & \cos\theta_x \end{pmatrix} \quad (2.2)$$

Realizando el producto de la matriz 2.2 ($R = R_z R_y R_x$):

$$R = \begin{pmatrix} \cos\theta_y \cos\theta_z & -\text{sen}\theta_z \cos\theta_x - \cos\theta_z \text{sen}\theta_y \text{sen}\theta_x & \text{sen}\theta_z \text{sen}\theta_x - \cos\theta_z \text{sen}\theta_y \cos\theta_x \\ \cos\theta_y \text{sen}\theta_z & \cos\theta_y \cos\theta_z - \text{sen}\theta_x \text{sen}\theta_y \text{sen}\theta_z & -\text{sen}\theta_x \cos\theta_z - \text{sen}\theta_y \text{sen}\theta_z \cos\theta_x \\ \text{sen}\theta_y & \cos\theta_y \text{sen}\theta_x & \cos\theta_y \cos\theta_x \end{pmatrix}$$

Suponiendo que los ángulos θ_i son pequeños (esta hipótesis es generalmente razonable si uno considera dos imágenes sucesivas obtenidas de la cámara de vídeo), uno efectúa un desarrollo limitado de primer orden de los términos trigonométricos (por consiguiente $\text{sen}\theta_i = \theta_i$, $\cos\theta_i = 1$ y $\theta_i\theta_j = 0$), uno obtiene la ecuación simplificada:

$$\begin{pmatrix} X_2 - X_{g1} \\ Y_2 - Y_{g1} \\ Z_2 - Z_{g1} \end{pmatrix} = \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} + \begin{pmatrix} 1 & -\theta_z & -\theta_y \\ \theta_z & 1 & -\theta_x \\ \theta_y & \theta_x & 1 \end{pmatrix} \begin{pmatrix} X_1 - X_{g1} \\ Y_1 - Y_{g1} \\ Z_1 - Z_{g1} \end{pmatrix} \quad (2.3)$$

Vectorialmente, uno también puede escribir las ecuaciones anteriores de la manera siguiente:

$$G_1 P_2 = \vec{T} + R_z \cdot R_y \cdot R_x \cdot \vec{G}_1 P_1 \quad (2.4)$$

o

$$P_1 P_2 = \vec{T} + P_1 \vec{G}_1 + R_z \cdot R_y \cdot R_x \cdot \vec{G}_1 P_1 \quad (2.5)$$

Sin embargo como los centros de rotación se toman de un centro de gravedad G (esta opción es sólo un problema de referencia, lo que no quita un carácter general de las ecuaciones siguientes), si $P_1 = G_1$ en un $P_2 = G_2$ y $\vec{T} = \vec{G}_1 \vec{G}_2$, ver figura 2.1. Uno puede escribir el vector de desplazamiento bajo la forma:

$$P_1 P_2 = \vec{P}_1 P_2 + R_z \cdot R_y \cdot R_x \cdot \vec{G}_1 P_1 \quad (2.6)$$

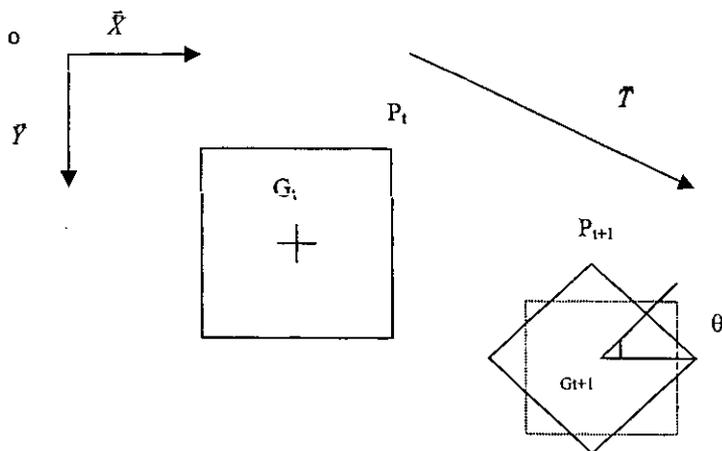


Figura 2.1. Descomposición de un movimiento.

Si uno considera las rotaciones alrededor de un eje paralelo al eje óptico Z [Anexo B], las ecuaciones se simplifican de la manera siguiente:

$$\begin{pmatrix} X_2 - X_1 \\ Y_2 - Y_1 \\ Z_2 - Z_1 \end{pmatrix} = \begin{pmatrix} X_{g2} - X_1 \\ Y_{g2} - Y_1 \\ Z_{g2} - Z_1 \end{pmatrix} + \begin{pmatrix} \cos\theta_z & -\text{sen}\theta_z & 0 \\ \text{sen}\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_1 - X_{g1} \\ Y_1 - Y_{g1} \\ Z_1 - Z_{g1} \end{pmatrix} \quad (2.7)$$

Las ecuaciones anteriores se usan para permitir unir movimientos 3D y 2D, vía una proyección del movimiento 3D en el plano imagen.

La utilización de un modelo de la cámara permite definir las relaciones geométricas existentes entre la posición de un punto en el espacio de tres dimensiones y su posición en el plano de la imagen. Si un punto $P(X, Y, Z)$ del espacio 3D es proyectado en el plano de la imagen en un punto $p(x, y)$ entonces:

$$x - x_0 = \frac{f}{px} \cdot \frac{X}{Z}$$

$$y - y_0 = \frac{f}{py} \cdot \frac{Y}{Z}$$

Donde las siguientes anotaciones son:

f : Distancia focal de la cámara.

x_0, y_0 : Coordenadas del centro óptico.

px y py Representan respectivamente la longitud en x y y de la cámara [13].

Para simplificar las anotaciones suponemos que $px = py = 1$, así tenemos:

$$x - x_0 = f \cdot \frac{X}{Z} \quad -$$

$$y - y_0 = f \cdot \frac{Y}{Z} \quad (2.8)$$

2.2 Modelo de movimiento 2D paramétrico

La opción de un modelo en el análisis del movimiento es fundamental porque a través del intermediario será posible adquirir un conocimiento global de sus movimientos presentados en una escena. Si uno desea transmitir la información de movimiento en una aplicación codificada, es necesario tener presente que uno intenta minimizar el volumen de información para transmitir. Para eso, es necesario tener modelos de movimiento utilizando un número mínimo de parámetros (de manera general, minimizar este volumen de información permite minimizar el costo operativo).

Hay dos modelos que resultan de dos aproximaciones fundamentales para la descripción del movimiento 2D:

(1) La proyección del movimiento 3D en el plano imagen: es un acercamiento geométrico.

(2) La aproximación lineal de un campo de vectores de velocidad 2D: es aproximación diferencial.

2.2.1 Proyección del movimiento 3D en el plano imagen

En esta parte se describe las relaciones analíticas que ligan coordenadas 2D y 3D de un objeto rígido en movimiento. Nuestro objetivo es definir descriptores (o parámetros) de movimiento aparente 2D representando directamente los posibles parámetros del movimiento 3D.

Si uno usa las ecuaciones (2.3) y (2.8) se obtienen las siguientes ecuaciones:

$$x_2 - x_1 = \frac{(Z_1 \cdot x_1 - x_1) + Z_{g2} \cdot x_{g2} - Z_{g1} \cdot x_{g1} - \theta_x (Z_1 y_1 - Z_{g1} y_{g1}) - \theta_y (Z_1 \cdot f - Z_{g1} \cdot f)}{Z_2}$$

$$y_2 - y_1 = \frac{(Z_1 \cdot y_1 - y_1) + Z_{g2} \cdot x_{g2} - Z_{g1} \cdot y_{g1} - \theta_z (Z_1 y_1 - Z_{g1} Y_{g1}) - \theta_y (Z_1 \cdot f - Z_{g1} \cdot f)}{Z_2}$$

$$Z_2 = Z_1 + Z_{g2} - Z_{g1} + \frac{\theta_y}{f} \cdot (Z_1 \cdot x_1 - Z_{g1} x_{g1}) + \frac{\theta_z}{f} \cdot (Z_1 \cdot y_1 - Z_{g1} y_{g1}) \quad (2.9)$$

El vector de desplazamiento es una relación de polinomio en x , y y z . Si uno no toma en cuenta los coeficientes del movimiento 3D; T_x , T_y , T_z y R_z (y anulando θ_x y θ_y), es posible obtener los parámetros simples del movimiento 2D; el movimiento de rotación alrededor de un eje paralelo al eje óptico (z) de la cámara y el centro de rotación en el centro de gravedad del objeto, éstas son las condiciones:

$$R = R_z = \begin{pmatrix} \cos \theta_z & -\text{sen} \theta_z & 0 \\ \text{sen} \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Desarrollando la ecuación (2.7) entre los instantes t_1 y t_2

$$X_2 - X_1 = X_{g2} - X_1 + \cos \theta_z (X_1 - X_{g1}) - \text{sen} \theta_z (Y_1 - Y_{g1})$$

$$Y_2 - Y_1 = Y_{g2} - Y_1 + \text{sen} \theta_z (X_1 - X_{g1}) + \cos \theta_z (Y_1 - Y_{g1})$$

$$Z_2 - Z_1 = Z_{g2} - Z_{g1} = T_z \quad (2.10)$$

Haciendo la proyección en el plano imagen:

$$x_2 - x_1 = \frac{Z_{g2}}{Z_2} (x_{g2} - x_0) - (x_1 - x_0) + \cos \theta_z \left(\frac{Z_1}{Z_2} (x_1 - x_0) - \frac{Z_{g1}}{Z_2} (x_{g1} - x_0) \right) - \text{sen} \theta_z \left(\frac{Z_1}{Z_2} (y_1 - y_0) - \frac{Z_{g1}}{Z_2} (y_{g1} - y_0) \right)$$

$$y_2 - y_1 = \frac{Z_{g2}}{Z_2}(y_{g2} - y_0) - (y_1 - y_0) + \text{sen } \theta_z \left(\frac{Z_1}{Z_2}(x_1 - x_0) - \frac{Z_{g1}}{Z_2}(x_{g1} - x_0) \right) \\ + \text{cos } \theta_z \left(\frac{Z_1}{Z_2}(y_1 - y_0) - \frac{Z_{g1}}{Z_2}(y_{g1} - y_0) \right)$$

Para simplificar las ecuaciones anteriores, se deben introducir tres hipótesis:

(1) Si uno considera el ángulo θ cero, es posible hacer un desarrollo de los términos de *seno* y *coseno* de 1er orden:

$$\text{sen } \theta = \theta \\ \text{cos } \theta = 1$$

Las ecuaciones anteriores se simplifican de la siguiente manera:

$$x_2 - x_1 = \frac{Z_{g2}}{Z_2}(x_{g2} - x_0) - (x_1 - x_0) + \frac{Z_1}{Z_2}(x_1 - x_0) - \frac{Z_{g1}}{Z_2}(x_{g1} - x_0) - \theta \left(\frac{Z_1}{Z_2}(y_1 - y_0) - \frac{Z_{g1}}{Z_2}(y_{g1} - y_0) \right) \\ y_2 - y_1 = \frac{Z_{g2}}{Z_2}(y_{g2} - y_0) - (y_1 - y_0) + \theta \left(\frac{Z_1}{Z_2}(x_1 - x_0) - \frac{Z_{g1}}{Z_2}(x_{g1} - x_0) \right) + \frac{Z_1}{Z_2}(y_1 - y_0) - \frac{Z_{g1}}{Z_2}(y_{g1} - y_0)$$

Eliminando términos.

$$x_2 - x_1 = \frac{Z_{g2}}{Z_2}(x_{g2} - x_{g1}) - \frac{T_z}{Z_2}(x_1 - x_{g1}) - \theta \left(\frac{Z_1}{Z_2}(y_1 - y_0) - \frac{Z_{g1}}{Z_2}(y_{g1} - y_0) \right) \\ y_2 - y_1 = \frac{Z_{g2}}{Z_2}(y_{g2} - y_{g1}) + \theta \left(\frac{Z_1}{Z_2}(x_1 - x_0) - \frac{Z_{g1}}{Z_2}(x_{g1} - x_0) \right) - \frac{T_z}{Z_2}(y_1 - y_{g1})$$

(2) A partir de las ecuaciones anteriores, es necesario modelar, para cada región, la superficie de sus objetos. Por lo tanto, el método más simple supone que la función $Z = f(X, Y)$ es una función lineal y se representa de la siguiente forma:

$$Z = Z_0 + aX + bY + o^2$$

La eliminación de términos de orden superior supone que los objetos de la escena tienen las superficies planas en cualquier orientación. Este modelo permite introducir tres nuevos parámetros para identificarlos. Para evitar estos inconvenientes, uno limita, la escena de este estudio, a un desarrollo limitado de orden cero. Por consiguiente, si $p(X_p, Y_p, Z_p)', q(X_p, Y_p, Z_p)' \in R$, entonces $Z_p = Z_q$.

Geoméricamente, esta hipótesis supone que cada "objeto" (o región) en movimiento puede descomponerse en un conjunto de superficies planas paralelas al plano imagen. Cada uno de estas superficies elementales forma una región [13].

En la práctica, esta hipótesis es determinante, porque los objetos (o imágenes) no siempre tendrán superficies planas que este modelo requiera. El error cometido, utilizando este modelo, se calcula de la siguiente manera:

Si tenemos que $Z_2 = Z_{g2} + \Delta Z_2$ y $Z_1 = Z_{g1} + \Delta Z_1$, son los puntos de la región, entonces:

$$x_2 - x_1 = \frac{Z_2 - \Delta \cdot Z_2}{Z_2} (x_{g2} - x_{g1}) - \frac{T_z}{Z_2} (x_1 - x_{g1}) - \theta \left(\frac{Z_1}{Z_2} (y_1 - y_0) - \frac{Z_1 - \Delta \cdot Z_1}{Z_2} (y_{g1} - y_0) \right)$$

$$y_2 - y_1 = \frac{Z_2 - \Delta \cdot Z_2}{Z_2} (y_{g2} - y_{g1}) - \frac{T_z}{Z_2} (y_1 - y_{g1}) + \theta \left(\frac{Z_1}{Z_2} (x_1 - x_0) - \frac{Z_1 - \Delta \cdot Z_1}{Z_2} (x_{g1} - x_0) \right)$$

Puede escribirse de la siguiente manera:

$$x_2 - x_1 = \left(1 - \frac{\Delta \cdot Z_2}{Z_2} \right) (x_{g2} - x_{g1}) - \frac{T_z}{Z_2} (x_1 - x_{g1}) - \theta \frac{Z_1}{Z_2} \left(y_1 - y_{g1} + \frac{\Delta \cdot Z_1}{Z_1} (y_{g1} - y_0) \right)$$

$$y_2 - y_1 = \left(1 - \frac{\Delta \cdot Z_2}{Z_2} \right) (y_{g2} - y_{g1}) - \frac{T_z}{Z_2} (y_1 - y_{g1}) + \theta \frac{Z_1}{Z_2} \left(x_1 - x_{g1} - \frac{\Delta \cdot Z_1}{Z_1} (x_{g1} - x_0) \right)$$

Si uno supone que para todo punto R es:

$$\frac{\Delta \cdot Z_2}{Z_2} \ll 1$$

$$\frac{\Delta \cdot Z_1}{Z_1} (x_{g1} - x_0) \ll x_1 - x_{g1}$$

$$\frac{\Delta \cdot Z_1}{Z_1} (y_{g1} - y_0) \ll y_1 - y_{g1}$$

Entonces la hipótesis de superficie plana paralela al plano imagen es válida, (esta hipótesis es generalmente válida si los objetos son alejados de la cámara).

Bajo estas hipótesis, se tiene entonces:

$$x_2 - x_1 = x_{g2} - x_{g1} - \frac{T_z}{Z_2} (x_1 - x_{g1}) - \theta \left(1 - \frac{T_z}{Z_2} \right) (y_1 - y_{g1})$$

$$y_2 - y_1 = y_{g2} - y_{g1} + \theta \left(1 - \frac{T_z}{Z_2} \right) (x_1 - x_{g1}) - \frac{T_z}{Z_2} (y_1 - y_{g1})$$

En lo siguiente, utilizaremos las anotaciones siguientes:

- $k = -\frac{T_z}{Z_2}$ Se llama "coeficiente de divergencia".
- $t_x = x_{g2} - x_{g1}$ y $t_y = y_{g2} - y_{g1}$ Se llaman "coeficientes de traslación".
- θ El ángulo de rotación alrededor del eje paralelo en el plano imagen que pasa por el centro de gravedad de la región considerada.

3) La tercera hipótesis se hace de las dos anteriores, consiste en suponer que los coeficientes de divergencia son suficientemente pequeños para eliminar los términos del segundo orden de la forma k y θ . Esta hipótesis es razonable según la

hipótesis (2) porque $k < 1$ y según la hipótesis (1) θ es pequeña. Se tiene entonces:

$$\begin{aligned} dx &= x_2 - x_1 = t_x + k(x_1 - x_{g1}) - \theta(y_1 - y_{g1}) \\ dy &= y_2 - y_1 = t_y + \theta(x_1 - x_{g1}) - k(y_1 - y_{g1}) \end{aligned} \quad (2.11)$$

2.2.2 Modelo de movimiento lineal simplificado

Si uno conoce (t_x, t_y, k, θ) en todas las regiones de la imagen, uno puede obtener el desplazamiento de cada píxel. Por consiguiente una descripción del movimiento aparente no depende de sus cuatro coeficientes "parámetros de movimiento" o "descriptores de movimiento". Un modelo es caracterizado a través de dos "ecuaciones de movimiento"; así:

$$\mathbf{d} = \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{pmatrix} k & -\theta \\ \theta & k \end{pmatrix} \cdot \begin{pmatrix} x - x_g \\ y - y_g \end{pmatrix} \quad (2.12)$$

su forma matricial con

$$M1 = \begin{pmatrix} k & -\theta \\ \theta & k \end{pmatrix} \quad (2.13)$$

de donde:

$$\mathbf{d} = \mathbf{T} + M1 \cdot GP \quad (2.14)$$

el vector de parámetros se escribe:

$$\theta_{ML} = \begin{pmatrix} t_x \\ t_y \\ k \\ \theta \end{pmatrix}$$

Se utiliza un centro de gravedad, "normalizado" (esto quiere decir que todos los términos son lineales), esto permite una interpretación más fácil de los coeficiente constantes. Este modelo tiene las siguientes siglas ML (para el modelo lineal).

La eliminación de algunos parámetros da tres modelos más simples. Respectivamente, si nosotros eliminamos k, θ o ambos parámetros respectivamente, nosotros obtenemos el modelo de divergencia (MD), el modelo de rotación (MR) y el modelo de movimiento constante (M0) [13].

Nota: Si entre dos instantes t_1 y t_2 la distancia focal de la cámara varía de $\Delta f = f_2 - f_1$ entonces cada punto $p(x_1, y_1, z_1)$ de la imagen se desplaza al punto $p(x_2, y_2, z_2)$ según las ecuaciones siguientes:

$$x_2 - x_o = f_2 \frac{X}{Z}$$

$$y_2 - y_o = f_2 \frac{Y}{Z}$$

$$x_1 - x_o = f_1 \frac{X}{Z}$$

$$y_1 - y_o = f_1 \frac{Y}{Z}$$

por consiguiente tenemos

$$dx = x_2 - x_1 = \frac{\Delta f}{f_1} (x_1 - x_o)$$

$$dy = y_2 - y_1 = \frac{\Delta f}{f_1} (y_1 - y_0)$$

El efecto en el movimiento aparente de una variación de distancia focal puede ser por consiguiente modelado de la misma manera como un movimiento de traslación de la cámara a lo largo del eje (mismo tipo de movimiento aparente 2D). El coeficiente de divergencia de k aloja por consiguiente una variación de distancia focal. La representación anterior del movimiento puede simplificarse teniendo en cuenta los parámetros del k y θ .

2.2.3 Modelo de movimiento constante

El modelo de movimiento constante tiene los términos t_x, t_y y elimina los términos de orden superior. Las ecuaciones del modelo se escriben:

$$\mathbf{d} = \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (2.15)$$

En el modelo, se toman movimientos 3D de traslación paralelo al plano imagen (T_x y T_y) es decir dos grados de libertad en seis. El vector de parámetros θ_{M0} es un vector de dos dimensiones.

$$\theta_{M0} = \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Este modelo tiene las siglas M0 (para el modelo constante).

Notas:

- En el caso donde la cámara está inmóvil, teóricamente donde $\theta_{M0} = 0$ corresponde a regiones (u objetos) estáticos, pero también puede corresponder a regiones en movimiento que contienen poca información textual (y por consiguiente no hay movimiento aparente identificable).
- En el caso de una estimación local de movimiento (píxel por píxel), el modelo de velocidades constantes es, por supuesto, el único modelo utilizable porque el desplazamiento de un punto siempre puede representarse solamente por dos coordenadas del desplazamiento del vector.
- El nombre de "modelo constante" no significa que el movimiento es constante en el tiempo, solamente que el movimiento de cada punto de la región es representado por el mismo vector de desplazamiento (esto se aplica en el caso de los modelos lineales).

2.2.4 Modelos de movimiento 2D de más de dos parámetros

A continuación se mostraran los siguientes modelos de movimiento en las tablas 2.1 y 2.2:

Modelo	Parámetro	Grado de libertad	Parámetros
Nulo		0	
Constante	M0	2	t_x, t_y
Divergencia	MD	3	t_x, t_y, k
Rotación	MR	3	t_x, t_y, θ
Lineal simplificado	ML	4	t_x, t_y, k, θ

Tabla.2.1. Ejemplos de modelos de movimiento y de variación de iluminación.

La figura 2.2 ilustra el fenómeno de rotación en el movimiento. Los ejes coordenados, además de trasladarse (t_x, t_y) rotan (θ). Por lo tanto, no es suficiente el modelo de movimiento

constante, dando origen a los modelos de movimiento de más de dos parámetros.

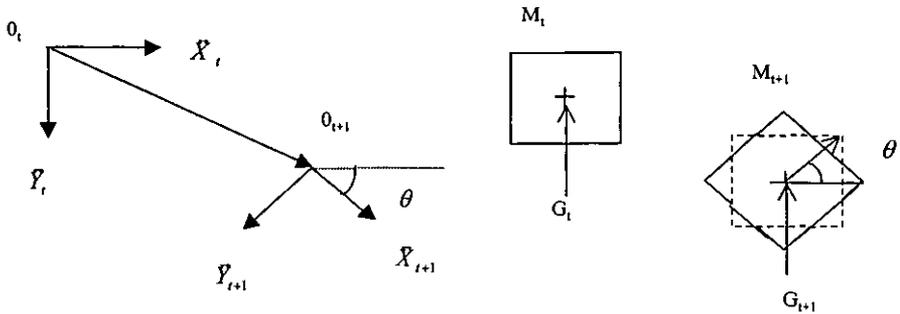


Figura 2.2. Movimiento relativo cámara-objeto ($\vec{M}, M_{r2} = d(dx, dy)$)

M0	$dx = t_x$ $dy = t_y$	$d = M_0$ $\theta_{M_0} = (t_x, t_y)^t$
MD	$dx = t_x + k(x - x_g)$ $dy = t_y + k(y - y_g)$	$d = M_0 + M_d \cdot GP$ $\theta_{MD}(t_x, t_y, k)^t$
MR	$dx = t_x - \theta(x - x_g)$ $dy = t_y + \theta(y - y_g)$	$d = M_0 + M_r \cdot GP$ $\theta_{MR}(t_x, t_y, \theta)^t$
ML	$dx = t_x + k(x - x_g) - \theta(y - y_g)$ $dy = t_y + k(y - y_g) + \theta(x - x_g)$	$d = M_0 + M_a \cdot GP$ $\theta_{ML}(t_x, t_y, k, \theta)^t$

Tabla 2.2. Ecuaciones de modelos de movimiento.

En donde

$$M_r = \begin{pmatrix} 0 & -\theta \\ \theta & 0 \end{pmatrix} \quad M_d = \begin{pmatrix} k & 0 \\ 0 & k \end{pmatrix}$$

2.3 Minimización basada en el gradiente del modelo 2D de cuatro parámetros

Los métodos pel-recursivo son estimadores del tipo predictor correctivo, de la forma,

$$\hat{\mathbf{d}}_a(\mathbf{x}, t; \Delta t) = \hat{\mathbf{d}}_b(\mathbf{x}, t; \Delta t) + \mathbf{u}(\mathbf{x}, t; \Delta t) \quad (2.16)$$

Donde $\hat{\mathbf{d}}_a(\mathbf{x}, t; \Delta t)$ denota el vector de movimiento estimado en la locación \mathbf{x} y el tiempo t , $\hat{\mathbf{d}}_b(\mathbf{x}, t; \Delta t)$ denota la estimación del movimiento predictivo, y $\mathbf{u}(\mathbf{x}, t; \Delta t)$ es el término de actualización. Los subíndices "a" y "b" denotan después y antes de la actualización en el pel (\mathbf{x}, t) localizado.

El estimador (2.16) es normalmente empleado de una manera recursiva, realizando uno o más iteraciones en (\mathbf{x}, t) y luego procediendo al próximo pixel en la dirección de la búsqueda. Generalmente, la mejor estimación en el anterior pel se tomó como la estimación para los próximos pel, seguidos por una sola actualización basada en el gradiente para minimizar la diferencia del cuadro desplazado en ese pel.

2.3.1 Diferencia del cuadro desplazado (DFD)

El principio fundamental en casi todos los métodos de estimación de movimiento, es conocido como la restricción del flujo óptico, es que la intensidad de la imagen permanece inalterada desde una secuencia a otra secuencia a lo largo del verdadero camino de movimiento (o cambios en una moda conocida o predecible). La restricción del flujo óptico puede emplearse en la forma de la ecuación de flujo óptico o puede ser impuesto minimizando la (diferencia del cuadro desplazado), como métodos block matching y pel-recursivo [2].

La DFD entre los casos de tiempo t y $t'=t+\Delta t$, se define por:

$$dfd(\mathbf{x}, \mathbf{d}) = s_c(\mathbf{x} + \mathbf{d}(\mathbf{x}, t; \Delta t), t + \Delta t) - s_c(\mathbf{x}, t) \quad (2.17)$$

Donde $s_c(x_1, x_2, t)$ denota la distribución de la imagen tiempo variante, y

$$\mathbf{d}(\mathbf{x}, t; \Delta t) = \mathbf{d}(\mathbf{x}) = [d_1(\mathbf{x}) \quad d_2(\mathbf{x})]^T$$

Denota el campo de vector de desplazamiento entre los tiempos t y $t'=t+\Delta t$. Observamos que

i) si las componentes de $\mathbf{d}(\mathbf{x})$ asume valores no enteros, se requiere interpolación [Anexo A], para computar la DFD a cada localización del píxel; y

ii) si $\mathbf{d}(\mathbf{x})$ fuera igual al verdadero vector de desplazamiento al sitio \mathbf{x} y no hubiera ningún error de la interpolación, el DFD logra el valor de cero a ese sitio bajo el obligado flujo óptico.

2.3.2 Optimización basada en el gradiente

El camino más recomendable para la minimización de una función $f(u_1, \dots, u_n)$ de varias incógnitas es calcular su parcial con respecto a cada una de estas incógnitas e igualarlas a cero y resolver las ecuaciones resultantes.

$$\frac{\partial f(\mathbf{u})}{\partial u_1} = 0$$

⋮

$$\frac{\partial f(\mathbf{u})}{\partial u_n} = 0$$

Este conjunto de ecuaciones simultaneas puede expresarse como un vector.

$$\nabla u f(\mathbf{u}) = 0 \quad (2.18)$$

Donde $\nabla \mathbf{u}$ es el operado gradiente con respecto al vector incógnita \mathbf{u} . Porque es difícil definir una función de criterio $f(u_1, \dots, u_n)$ para estimar el movimiento y/o resolver el conjunto de ecuaciones (2.18) en forma cerrada, tenemos que recurrir a métodos iterativos (métodos numéricos). Por ejemplo la DFD es una función de intensidad de píxel que no puede ser expresada en forma cerrada.

Método del descenso a pasos

Esté método es probablemente el método de optimización más simple. Actualiza el valor del dato estimado del mínimo local en la dirección del gradiente negativo. Recordando que el vector gradiente apunta en la dirección del máximo. Esto es, en una dimensión (función de una sola variable), su signo será positivo en una pendiente cuesta arriba. Por esta razón, la dirección del descenso inclinado es justamente la dirección opuesta, la cual se ilustra en la figura .2.3.

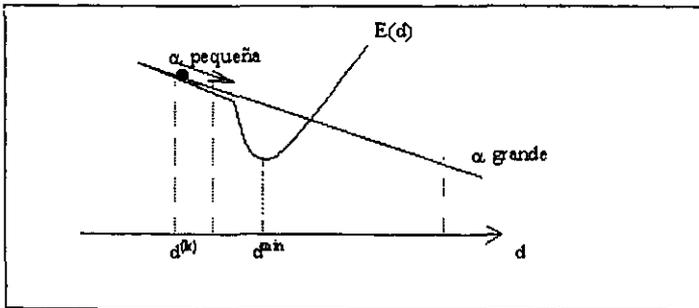


Figura 2.3. Una ilustración de la método del gradiente descendente.

Por lo tanto, para acercarnos al mínimo, actualizamos nuestro valor estimado de la forma:

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \alpha \nabla \mathbf{u} f(\mathbf{u}) \Big|_{\mathbf{u}^{(k)}}$$

Donde α es un escalar positivo, conocido como el tamaño del paso. El tamaño del paso es determinante para que las iteraciones convergan, porque si α es muy pequeño nos acercamos muy lentamente y las iteraciones tardarían mucho para converger. Por otro lado, si lo tomamos muy grande el algoritmo podría llegar a convertirse en inestable de tal manera que podría estar oscilando cerca del mínimo pero nunca llegar a él. En el método del descenso a pasos, el tamaño del paso es cambiado heurísticamente.

Método Newton-Raphson

Para optimizar el tamaño del paso α podría ser estimado usando el método muy conocido de Newton-Raphson para encontrar raíces. Para el caso de la derivada de una función de una sola variable es muy simple. En una dimensión, nos gustaría encontrar la raíz de $f'(u)$. Para esto, debemos expandir a $f'(u)$ en series de Taylor en el punto $u^{(k)}$ para obtener

$$f'(u^{(k+1)}) = f'(u^{(k)}) + (u^{(k+1)} - u^{(k)})f''(u^{(k)})$$

Porque el deseo es que $f'(u^{(k+1)})$ sea cero, se tiene:

$$f'(u^{(k)}) + (u^{(k+1)} - u^{(k)})f''(u^{(k)}) = 0 \quad (2.19)$$

Resolviendo (2.19) para $u^{(k+1)}$, tenemos

$$u^{(k+1)} = u^{(k)} - \frac{f'(u^{(k)})}{f''(u^{(k)})}$$

Este resultado puede ser generalizado para el caso de una función de varias incógnitas como

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \mathbf{H}^{-1} \nabla \mathbf{u} f(\mathbf{u}) \Big|_{\mathbf{u}^{(k)}}$$

Donde \mathbf{H} es la matriz Hessiana

$$\mathbf{H}_{ij} = \left[\frac{\partial^2 f(\mathbf{u})}{\partial u_i \partial u_j} \right]$$

El método de Newton-Raphson encuentra una expresión analítica para el tamaño del paso en términos de la segunda parcial de la función. Cuando no se encuentra una forma cerrada de la función, la matriz Hessiana puede ser estimada usando métodos numéricos.

Mínimo local vs. global

La aproximación por el descenso del gradiente sufre de algunos inconvenientes: la solución depende del punto inicial. Si nosotros empezamos en un valle éste se pegara al fondo de ese valle, aun si es un mínimo local. Ya que el gradiente será cero o casi cero alrededor de un mínimo local. Una solución a este problema es escoger varios puntos de comienzo, y elegir la solución más optima.

Existen en la literatura métodos de optimización más sofisticados pero requieren más tiempo de proceso.

2.3.3 Algoritmos basados en el método “Descenso a pasos”

La estimación de movimiento Pel-recursiva es usualmente precedida por una etapa de detección de cambio, donde la diferencia entre secuencias en cada píxel es probada contra un umbral. La estimación es realizada sólo en estos píxeles que pertenecen a la región cambiada. Estos algoritmos, basados en el método “descenso a pasos” pel-recursivo, estiman el termino actual $\mathbf{u}(\mathbf{x}, t; \Delta t)$ de la ecuación (2.16) en cada píxel en la región cambiada, al minimizar una función de criterio E de la DFD con respecto a \mathbf{d} .

Algoritmo de Netravali-Robbins

El algoritmo de Netravali-Robbins encuentra una estimación del vector de desplazamiento, el cual minimiza el cuadrado de la DFD en cada píxel, usando un método del descenso del gradiente. Entonces la función es minimizada como

$$E(\mathbf{x};\mathbf{d}) = [dfd(\mathbf{x}, \mathbf{d})]^2$$

La minimización de $E(\mathbf{x};\mathbf{d})$ con respecto a \mathbf{d} , en el píxel \mathbf{x} , por el método de "descenso a pasos" produce la iteración

$$\begin{aligned} \mathbf{d}^{i+1}(\mathbf{x}) &= \mathbf{d}^i(\mathbf{x}) - (1/2)\varepsilon \nabla_{\mathbf{d}} [dfd(\mathbf{x}, \mathbf{d}) \mid \mathbf{d}=\mathbf{d}^i]^2 \\ &= \mathbf{d}^i(\mathbf{x}) - \varepsilon dfd(\mathbf{x}, \mathbf{d}^i) \nabla_{\mathbf{d}} dfd(\mathbf{x}, \mathbf{d}) \mid \mathbf{d}=\mathbf{d}^i \end{aligned}$$

Donde ∇ es el gradiente con respecto a \mathbf{d} , y ε es el tamaño del paso. Recordando que el gradiente apunta a la dirección negativa en el método "descenso inclinado".

Algoritmo de Walker-Rao

Walker y Rao propusieron un método adaptado para obtener el tamaño del paso.

$$\varepsilon = \frac{1}{2 \|\nabla I^{t-1}(p + \mathbf{d})\|}$$

Ceffario y Rocca también tuvieron un desarrollo similar para la expresión del tamaño del paso.

$$\varepsilon = \frac{1}{2 \|\nabla I^{t-1}(p + \mathbf{d})\| + \eta^2}$$

El cual incluye un término η^2 que prevé la división entre cero en áreas de intensidad constante y el gradiente se hace casi cero. Experimentalmente, los resultados indicaban que usando la adaptación del tamaño del paso mejoraba la convergencia del

algoritmo de Netralval-Robbins. Se encontraba que con solo cinco iteraciones era suficiente para llegar a un resultado satisfactorio [2].

Algoritmo a utilizar

El método llamado '**método del gradiente**' es uno de los métodos usuales en teoría de la optimización cuando uno quiere minimizar una función de energía. Por ejemplo, en el método pel-recursivo se determina la dirección del gradiente para llegar a estimar el desplazamiento de manera iterativa usando la ecuación siguiente:

$$\mathbf{d}^{i+1} = \mathbf{d}^i - \frac{\varepsilon}{2} \mathbf{G}^i \quad (2.20)$$

con

$$\mathbf{G}^i = \begin{pmatrix} \frac{\partial}{\partial dx} DFD^2(x, y, \mathbf{d}^i) \\ \frac{\partial}{\partial dy} DFD^2(x, y, \mathbf{d}^i) \end{pmatrix} \quad (2.21)$$

Anotaciones:

i representan la i -ésima iteración.

\mathbf{G}^i es el término correctivo a la iteración i .

ε es una matriz de ganancia diagonal (2x2).

Es posible generalizar esta técnica estimando directamente el desplazamiento \mathbf{d} píxel por píxel, por el vector de desplazamiento $\Theta_{j,\beta}$ región por región [14]. En la descripción uno utiliza para cada región $R_{j,\beta}$ la ecuación iterativa siguiente:

$$\Theta_{j,\beta}^{i+1} = \Theta_{j,\beta}^i - \frac{\varepsilon_\beta}{2.N_j} \mathbf{G}^i \quad (2.22)$$

Una restricción de conservación de intensidad constante puede plantearse como

$$I^i(\mathbf{p}) = I^{i-1}(\mathbf{p} + \mathbf{d}) \quad (2.23)$$

en donde

$\mathbf{p} = (x, y)$ vector de coordenadas

$\mathbf{d} = (dx, dy)$ vector de desplazamiento

$dx = tx + k(x - xg) - \theta(y - yg)$ desplazamiento en la dirección x .

$dy = ty + k(y - yg) + \theta(x - xg)$ desplazamiento en la dirección y .

(xg, yg) centro de gravedad de la región R_i .

$\Theta = (tx, ty, k, \theta)$ vector de parámetros.

La ecuación (2.23) se puede resolver para \mathbf{d} mediante la minimización del término de conservación de intensidad constante

$$ED(\mathbf{p}, \mathbf{d}) = \rho(I^i(\mathbf{p}) - I^{i-1}(\mathbf{p} + \mathbf{d})) \quad (2.24)$$

$$ED(\mathbf{p}, \mathbf{d}) = \rho(DFD(\mathbf{p}, \mathbf{d})) \quad (2.25)$$

en donde

$DFD(\mathbf{p}, \mathbf{d}) = I^i(\mathbf{p}) - I^{i+1}(\mathbf{p} + \mathbf{d})$ Diferencia de cuadro desplazado.

$\rho(z)$ Estimador

Cuando $\rho(z) = z^2$, corresponde al estimador estándar por mínimos cuadrados.

Resolviendo la minimización de la ecuación (2.25) por el método del gradiente, nos conduce al cálculo interactivo del vector de parámetros.

$$\Theta_{j,\beta}^{i+1} = \Theta_{j,\beta}^i - \frac{1}{2N_j} \sum_{p \in R_j} \epsilon_\beta \mathbf{G}^i \quad (2.26)$$

en donde

N_j Tamaño de la región R_j .

ϵ_β Matriz de ganancia adaptable.

$$\epsilon_\beta = \frac{1}{\nabla_x^2 f(\mathbf{p} + \mathbf{d}_{j,\beta}, t_1) + \nabla_y^2 f(\mathbf{p} + \mathbf{d}_{j,\beta}, t_1) + \alpha} \begin{bmatrix} \epsilon_c & 0 & 0 & 0 \\ 0 & \epsilon_c & 0 & 0 \\ 0 & 0 & \epsilon_l & 0 \\ 0 & 0 & 0 & \epsilon_l \end{bmatrix} \quad (2.27)$$

en donde

$$\mathbf{G}^i = \begin{bmatrix} \frac{\partial}{\partial tx} ED(\mathbf{p}, \mathbf{d}) \\ \frac{\partial}{\partial ty} ED(\mathbf{p}, \mathbf{d}) \\ \frac{\partial}{\partial k} ED(\mathbf{p}, \mathbf{d}) \\ \frac{\partial}{\partial \theta} ED(\mathbf{p}, \mathbf{d}) \end{bmatrix} \quad (2.28)$$

$$\mathbf{G}^i = \begin{bmatrix} \frac{\partial}{\partial tx} ED(\mathbf{p}, \mathbf{d}) \\ \frac{\partial}{\partial ty} ED(\mathbf{p}, \mathbf{d}) \\ \frac{\partial}{\partial k} ED(\mathbf{p}, \mathbf{d}) \\ \frac{\partial}{\partial \theta} ED(\mathbf{p}, \mathbf{d}) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial tx} \rho(DFD(\mathbf{p}, \mathbf{d})) \\ \frac{\partial}{\partial ty} \rho(DFD(\mathbf{p}, \mathbf{d})) \\ \frac{\partial}{\partial k} \rho(DFD(\mathbf{p}, \mathbf{d})) \\ \frac{\partial}{\partial \theta} \rho(DFD(\mathbf{p}, \mathbf{d})) \end{bmatrix} \quad (2.29)$$

sabemos que $DFD(\mathbf{p}, \mathbf{d}) = I^t(\mathbf{p}) - I^{t+1}(\mathbf{p} + \mathbf{d})$. desarrollando el estimador estándar por mínimos cuadrados [14].

$\rho(\text{estimador})$

$$\rho(z) = z^2$$

sustituyendo el estimador

$$\rho(DFD) = DFD^2$$

Desarrollando la parcial de DFD^2 tenemos que:

$$\frac{\partial DFD^2}{\partial t_i}(\mathbf{p}, \mathbf{d}) = 2DFD(\mathbf{p}, \mathbf{d}) \cdot \frac{\partial DFD(\mathbf{p}, \mathbf{d})}{\partial t_i}$$

Por definición del DFD uno tiene:

$$\frac{\partial DFD(\mathbf{p}, \mathbf{d})}{\partial t_i} = - \frac{\partial I(\mathbf{p}, \mathbf{d})}{\partial t_i}$$

de donde:

$$\frac{\partial}{\partial t_i} DFD(\mathbf{p}, \mathbf{d}) = - \left[\frac{\partial}{\partial x} I^{i-1}(\mathbf{p}, \mathbf{d}) \cdot \frac{\partial}{\partial t_i} (x - dx) + \frac{\partial}{\partial y} I^{i-1}(\mathbf{p}, \mathbf{d}) \cdot \frac{\partial}{\partial t_i} (y - dy) \right]$$

entonces tenemos, sustituyendo en la parcial tenemos:

$$\frac{\partial DFD^2(\mathbf{p}, \mathbf{d})}{\partial t_i} = 2DFD(\mathbf{p}, \mathbf{d}) \cdot - \left[\frac{\partial}{\partial x} I^{i-1}(\mathbf{p}, \mathbf{d}) \cdot \frac{\partial}{\partial t_i} (x - dx) + \frac{\partial}{\partial y} I^{i-1}(\mathbf{p}, \mathbf{d}) \cdot \frac{\partial}{\partial t_i} (y - dy) \right]$$

$$i_1 = tx$$

Desarrollando para $i=4$;

$$i_2 = ty$$

$$i_3 = k$$

$$i_4 = \theta$$

Sustituyendo en G^i

$$\mathbf{G}^i = -2DFD(\mathbf{p}, \mathbf{d}) \begin{bmatrix} \left[\frac{\partial}{\partial x} I^{i-1}(\mathbf{p}, \mathbf{d}) \cdot \frac{\partial}{\partial tx} (x - dx) + \frac{\partial}{\partial y} I^{i-1}(\mathbf{p}, \mathbf{d}) \cdot \frac{\partial}{\partial tx} (y - dy) \right] \\ \left[\frac{\partial}{\partial x} I^{i-1}(\mathbf{p}, \mathbf{d}) \cdot \frac{\partial}{\partial ty} (x - dx) + \frac{\partial}{\partial y} I^{i-1}(\mathbf{p}, \mathbf{d}) \cdot \frac{\partial}{\partial ty} (y - dy) \right] \\ \left[\frac{\partial}{\partial x} I^{i-1}(\mathbf{p}, \mathbf{d}) \cdot \frac{\partial}{\partial k} (x - dx) + \frac{\partial}{\partial y} I^{i-1}(\mathbf{p}, \mathbf{d}) \cdot \frac{\partial}{\partial k} (y - dy) \right] \\ \left[\frac{\partial}{\partial x} I^{i-1}(\mathbf{p}, \mathbf{d}) \cdot \frac{\partial}{\partial \theta} (x - dx) + \frac{\partial}{\partial y} I^{i-1}(\mathbf{p}, \mathbf{d}) \cdot \frac{\partial}{\partial \theta} (y - dy) \right] \end{bmatrix} \quad (2.30)$$

tenemos que:

$$\begin{aligned} dx &= t_x + k(x - x_g) - \theta(y - y_g) \\ dy &= t_y + k(y - y_g) + \theta(x - x_g) \end{aligned}$$

Resolviendo las derivadas parciales:

$$\frac{\partial}{\partial tx} (x - dx) = \frac{\partial}{\partial tx} (x - tx - k(x - x_g) + \theta(y - y_g)) = -1$$

$$\frac{\partial}{\partial tx} (y - dy) = \frac{\partial}{\partial tx} (y - ty - k(y - y_g) - \theta(x - x_g)) = 0$$

$$\frac{\partial}{\partial ty} (x - dx) = \frac{\partial}{\partial ty} (x - tx - k(x - x_g) + \theta(y - y_g)) = 0$$

$$\frac{\partial}{\partial ty} (y - dy) = \frac{\partial}{\partial ty} (y - ty - k(y - y_g) - \theta(x - x_g)) = -1$$

$$\frac{\partial}{\partial k} (x - dx) = \frac{\partial}{\partial k} (x - tx - k(x - x_g) + \theta(y - y_g)) = -(x - x_g)$$

$$\frac{\partial}{\partial k} (y - dy) = \frac{\partial}{\partial k} (y - ty - k(y - y_g) - \theta(x - x_g)) = -(y - y_g)$$

$$\frac{\partial}{\partial \theta}(x - dx) = \frac{\partial}{\partial \theta}(x - tx - k(x - xg) + \theta(y - yg)) = (y - yg)$$

$$\frac{\partial}{\partial \theta}(y - dy) = \frac{\partial}{\partial \theta}(y - ty - k(y - yg) - \theta(x - xg)) = -(x - xg)$$

Sin embargo cuando el desplazamiento del vector está en función de parámetros de ecuaciones del movimiento, es posible expresar explícitamente \mathbf{G} .

Por definición del gradiente sabemos que:

$$\begin{aligned} \frac{\partial}{\partial x} I^{t-1}(\mathbf{p}, \mathbf{d}) &= \nabla_x I^{t-1}(\mathbf{p}, \mathbf{d}) \\ \frac{\partial}{\partial y} I^{t-1}(\mathbf{p}, \mathbf{d}) &= \nabla_y I^{t-1}(\mathbf{p}, \mathbf{d}) \end{aligned}$$

Sustituyendo las ecuaciones anteriores en la ecuación (2.30).

$$\mathbf{G}^i = -2DFD(\mathbf{p}, \mathbf{d}) \begin{bmatrix} -\nabla_x I^{t-1}(\mathbf{p}, \mathbf{d}) \\ -\nabla_y I^{t-1}(\mathbf{p}, \mathbf{d}) \\ -\nabla_x I^{t-1}(\mathbf{p}, \mathbf{d})(x - xg) - \nabla_y I^{t-1}(\mathbf{p}, \mathbf{d})(y - yg) \\ \nabla_x I^{t-1}(\mathbf{p}, \mathbf{d})(y - yg) - \nabla_y I^{t-1}(\mathbf{p}, \mathbf{d})(x - xg) \end{bmatrix} \quad (2.31)$$

Finalmente nos queda:

$$\Theta_{j,\beta}^{i+1} = \Theta_{j,\beta}^i - \frac{1}{2N_j} \sum_{p \in R_j} \varepsilon_\beta \mathbf{G}^i$$

$$\begin{bmatrix} tx^{i+1} \\ ty^{i+1} \\ k^{i+1} \\ \theta^{i+1} \end{bmatrix} = \begin{bmatrix} tx^i \\ ty^i \\ k^i \\ \theta^i \end{bmatrix} - \frac{1}{2N_j} \sum_{p \in R_j} \varepsilon_\beta 2DFD(\mathbf{p}, \mathbf{d}) \begin{bmatrix} -\nabla_x I^{t-1}(\mathbf{p}, \mathbf{d}) \\ -\nabla_y I^{t-1}(\mathbf{p}, \mathbf{d}) \\ -\nabla_x I^{t-1}(\mathbf{p}, \mathbf{d})(x - xg) - \nabla_y I^{t-1}(\mathbf{p}, \mathbf{d})(y - yg) \\ \nabla_x I^{t-1}(\mathbf{p}, \mathbf{d})(y - yg) - \nabla_y I^{t-1}(\mathbf{p}, \mathbf{d})(x - xg) \end{bmatrix}$$

$$\begin{bmatrix} tx^{i+1} \\ ty^{i+1} \\ k^{i+1} \\ \theta^{i+1} \end{bmatrix} = \begin{bmatrix} tx^i \\ ty^i \\ k^i \\ \theta^i \end{bmatrix} - \frac{1}{N_j} \sum_{p \in \beta_j} \frac{DFD(\mathbf{p}, \mathbf{d})}{[\nabla^2 xI^{i-1}(\mathbf{p}, \mathbf{d}) + \nabla^2 yI^{i-1}(\mathbf{p}, \mathbf{d}) + \alpha]} \bullet$$

$$\bullet \begin{bmatrix} \varepsilon_c \nabla xI^{i-1}(\mathbf{p}, \mathbf{d}) \\ \varepsilon_c \nabla yI^{i-1}(\mathbf{p}, \mathbf{d}) \\ \varepsilon_l (x - xg) \nabla xI^{i-1}(\mathbf{p}, \mathbf{d}) + \varepsilon_l (y - yg) \nabla yI^{i-1}(\mathbf{p}, \mathbf{d}) \\ -\varepsilon_l (y - yg) \nabla xI^{i-1}(\mathbf{p}, \mathbf{d}) + \varepsilon_l (x - xg) \nabla yI^{i-1}(\mathbf{p}, \mathbf{d}) \end{bmatrix}$$

La matriz de ganancia ε_{β} es:

$$\varepsilon_{\beta 0} = \begin{pmatrix} \varepsilon_c & 0 & . & . & . & 0 \\ 0 & \varepsilon_c & . & . & . & . \\ . & . & \varepsilon_l & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & 0 \\ 0 & . & . & . & 0 & \varepsilon_l \end{pmatrix}$$

donde $\varepsilon_c \gg \varepsilon_l$.

Por otro lado, los gradientes son filtros 3x3 (respectivamente para el gradiente en x y en y).

$$\nabla_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

$$\nabla_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

El algoritmo que implementa la ecuación (2.26), se muestra en la figura 2.4. Parte del código que se implementó, se muestra en el [Anexo C].

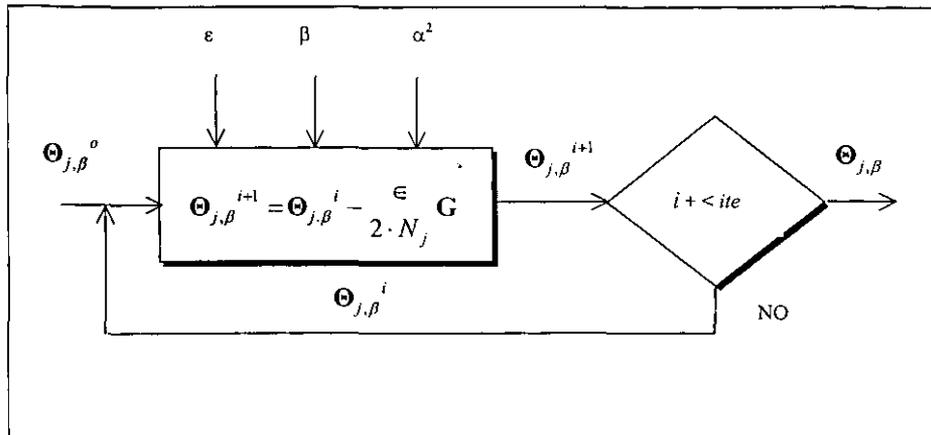


Figura 2.4. Esquema general del método del gradiente.

Para cada región de j de la imagen

Inicializando $\{\Theta_{j,\beta}^0\}_{j=1..N}$

Si $i <$ iteración

Si $DFD(\mathbf{p}, \mathbf{d}_{estimada}^{i+1}) < DFD(\mathbf{p}, \mathbf{d}_{j,\beta}^i)$

$\Theta_{j,\beta}^{i+1} := \Theta_{estimada}^{i+1}$

fin si

$i := i + 1$

fin.

CAPÍTULO 3

Visión estéreo

La poderosa herramienta de visión estéreo nos ayuda en diferentes ámbitos de nuestra vida diaria. Los humanos estamos dotados de esta herramienta que está incluida en nuestro sistema visual pero pocas veces nos preguntamos cual es el proceso que nos ayuda a “distinguir” el mundo real. En el siguiente capítulo explicaremos el método y proceso utilizado para tener un sistema que nos ayude establecer un sistema global a partir de dos proyecciones.

3.1. Conceptos básicos

Se han intentado varias tecnologías 3D que han encontrado demanda para ambientes realistas desde que la estereoscopia se desarrolló en los años 1800's. Varios acercamientos han fallado, incluso el estéreo y la holografía, desde el punto de vista de la aplicación práctica. Por ejemplo, aunque los métodos estereofónicos son una de las soluciones simples, mantener la percepción 3D en el despliegue había fallado porque no pudieron proporcionar una calidad alta o eliminar la incomodidad de las lentes especiales. En los años 90's, sin embargo, los métodos del estereoscopia recobraron el interés en las tecnologías 3D con el desarrollo de despliegue de la estereoscopia así como las

innovaciones en el campo electro-ópticas, telecomunicaciones, y sistemas de la computadora.

Las técnicas de visión estéreo son una manera simple de proporcionar señales de profundidad, las cuales dan como resultado el realismo esperado. La percepción de profundidad basada en técnicas de visión estéreo es obtenida viendo una escena simultáneamente en posiciones diferentes. En el sistema visual humano, cada ojo captura su propia vista y la información de profundidad se pierde desde que el mundo real se proyecta hacia la retina como imágenes 2D. Entonces, las dos imágenes separadas se envían al cerebro para su proceso. Cuando las dos imágenes llegan simultáneamente a la corteza visual primaria, están unidos en un cuadro. La mente (o un conocimiento anterior en el mundo 3D) combina las dos imágenes apareándose las similitudes y agregando las diferencias pequeñas. Las diferencias pequeñas entre las dos imágenes suman a una diferencia grande en el cuadro final. La imagen combinada es más que la suma de sus partes. Es un cuadro estéreo 3D, ver figura 3.1.

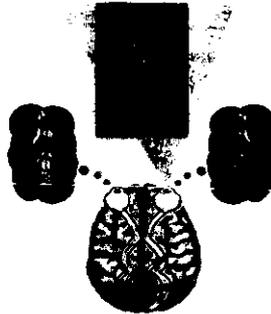


Figura 3.1. Percepción en 3D.

Como se verá a continuación, si nosotros podemos encontrar los puntos correspondientes en las imágenes estéreo, entonces podemos reconstruir estructuras 3D por una triangulación con disparidad y una geometría o (modelo) de la cámara predeterminada. Esto es conocido como el problema de la correspondencia.

Fundamentos

La idea en este caso es bastante similar al de estimación de movimiento, excepto que en vez de utilizar imágenes separadas entre sí por el tiempo, se utilizan dos (o más imágenes) separadas entre sí por un espacio, como sucede en los ojos de los humanos. Como un determinado aspecto de la escena se encuentra en un sitio diferente en relación con el eje z de cada plano de imagen, al sobreponer las dos imágenes habrá una disparidad en cuanto a la ubicación de la imagen original en las dos imágenes producidas. Lo anterior se puede ver muy claramente en la figura 3.2 en donde el punto más próximo de la pirámide aparece desplazado a la izquierda de la imagen derecha y a la derecha de la imagen izquierda.

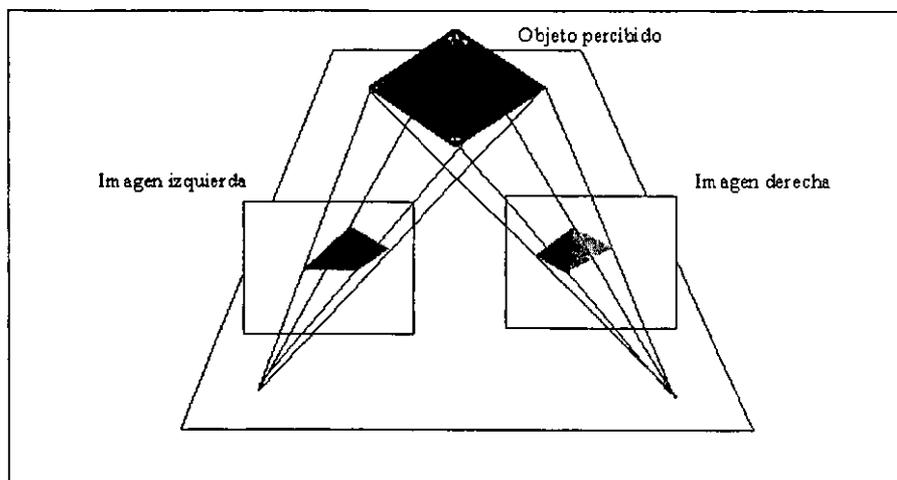


Figura 3.2. La idea de la visión estereoscópica: en cada posición de la cámara se obtiene una perspectiva 2D de la misma escena 3D.

- En condiciones de visión normal, los humanos efectúan una fijación; es decir, en la escena hay un punto en el que los ejes ópticos de los ojos se cruzan. En la figura 3.3 se muestran dos ojos que miran hacia el punto P_o , el que se encuentra a una distancia Z del punto que está entre ambos ojos. Conviene calcular la disparidad angular en radianes. La disparidad en el punto de fijación P_o es cero. Es posible calcular el desplazamiento angular

respecto de otro punto P de la escena, alejado δZ de las imágenes izquierda y derecha, desplazamientos a los que denominaremos P_L y P_R . Si cada uno de los anteriores se desplaza en relación con P es de sólo $\delta\theta$. Utilizando simple geometría tenemos que:

$$\frac{\delta\theta}{\delta Z} = \frac{-b}{Z^2}$$

En el caso de los humanos. b (la línea de base) es de aproximadamente 6 cm. Suponga que Z mide aproximadamente 100 cm. La mínima $\delta\theta$ detectable (que corresponde al tamaño de un píxel) es de aproximadamente 5 segundos de arco, o 2.42×10^{-5} radianes, lo que produce una δZ de aproximadamente 0.4 mm. Si $Z = 30$ cm. (1 pie), se obtiene un valor impresionantemente pequeño $\delta Z = 0.036$ mm. Es decir, a una distancia de 30 cm los seres humanos son capaces de distinguir profundidades que difieren entre sí tan sólo en 0.036 mm, lo que nos permite realizar tareas como la de pasar un hilo por el ojo de una aguja.

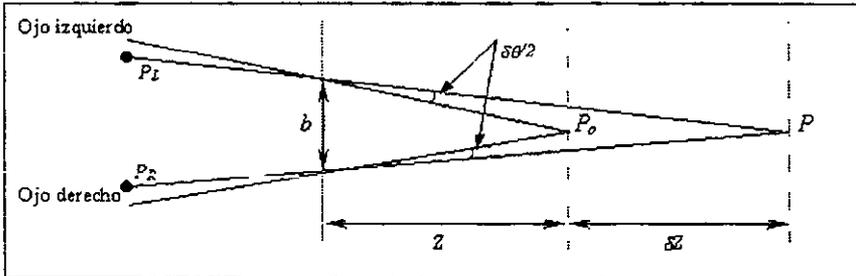


Figura 3.3. La relación entre disparidad y profundidad en la visión estereoscópica.

A diferencia del movimiento, se supone que conocemos la geometría de la visión o la orientación relativa entre los ojos, supuesto que es razonable [4]. En el caso de los ojos, el cerebro ordena un determinado estado a los músculos oculares para que desplacen los ojos, lo que permite conocer la posición de cada ojo en relación con la cabeza. De igual manera, en un sistema de cámara binocular se sabe cuál es la configuración relativa.

El conocer la geometría de la vista es muy útil para medir la disparidad. Como en el caso del flujo óptico, podemos tratar de

localizar puntos entre las imágenes izquierda y derecha maximizando una determinada medida de similitud. Sin embargo, la búsqueda no tiene que realizarse en una región bidimensional. Los puntos de correspondencia siempre se encuentran a lo largo de líneas epipolares de las imágenes (véase figura 3.4). estas líneas corresponden a las intersecciones de un plano epipolar (el plano a través de un punto de la escena y los puntos nodales de ambos ojos) y de los planos de imágenes izquierdo y derecho. El aprovechamiento de esta restricción epipolar reduce una búsqueda originalmente bidimensional a una de tipo unidimensional. Es obvio que para definir las líneas epipolares se necesita conocer la geometría de la vista.

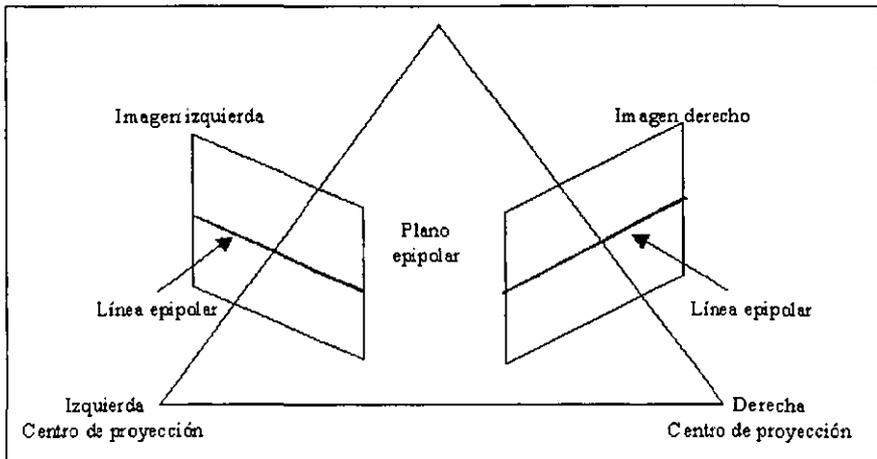


Figura 3.4. Imagen epipolar.

Un procedimiento ingenuo para localizar la disparidad consistiría en buscar a través de líneas epipolares, tratando de maximizar la intercorrelación, como en el caso del flujo óptico. Dado un punto p_i de la vista izquierda, para obtener el punto respectivo q_i de la vista derecha se busca a través de la correspondiente línea epipolar de otra vista. Se calcula la intercorrelación entre ventanas centradas en p_i y q para cada correspondencia posible q . El punto que corresponda se considera como el píxel q_i para el que se maximiza la intercorrelación.

Es conveniente aprovechar alguna de las siguientes restricciones:

- **Unicidad:** el punto de una imagen corresponde cuando más a un punto de la otra imagen. Decimos cuando más uno ya que es posible que el punto aparezca como incluido en otra vista.
- **La continuidad pieza a pieza de las superficies de una escena:** el hecho de que el mundo normalmente muestra continuidad de una pieza a otra implica que en los puntos de una escena cercanos entre sí los valores de profundidad también se asemejan, así como también los de disparidad, excepto en los límites y oclusiones de los objetos.

Fotografía y la transformación de la perspectiva

Una fotografía es una proyección. El caso general de una cámara se ilustra en la figura 3.5. El centro de la proyección es el punto focal de la lente de la cámara. Es conveniente considerar la creación del negativo de la fotografía original y de una impresión de ese negativo como dos casos separados.

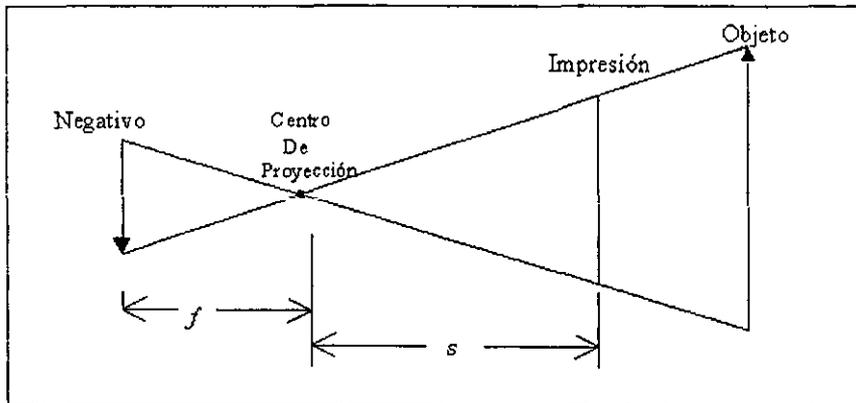


Figura 3.5. Una fotografía como una proyección de perspectiva.

El la figura 3.6 se ilustra la geometría para la creación del negativo original. Aquí es conveniente colocar el negativo en el plano $z = 0$ con centro de proyección y la escena localizadas en la mitad del espacio $z < 0$. La proyección de perspectiva en el plano $z = 0$ (negativo) se obtiene la transformación:

$$[T_n] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Donde f es el foco de la lente. Note que una imagen invertida del objeto es formada en el negativo. Específicamente,

$$[x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/f \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x \ y \ 0 \ 1+z/f]$$

Y

$$x^* = \frac{xf}{f+z} \quad y^* = \frac{yf}{f+z}$$

Aquí, para $f+z < 0$, x^* y y^* son de signo opuesto para x y y , y las imágenes inversas formadas en el negativo [15].

La figura 3.6(b) ilustra la geometría para la creación de una impresión del negativo de la fotografía. Aquí s es la distancia desde el punto focal de la lente al papel. El papel se asume que está localizado en $z=0$. la transformación de la proyección de perspectiva es entonces:

$$[T_p] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/s \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

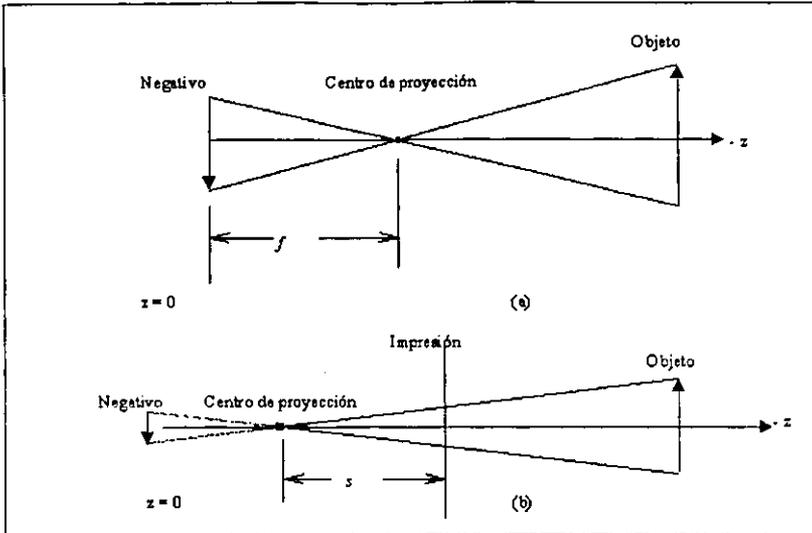


Figura 3.6. Geometría de una fotografía en perspectiva.

Note que en una imagen vertical del objeto se forma en el área de impresión [15]. Específicamente,

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/s \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 - \frac{z}{s} \end{bmatrix}$$

Y

$$x^* = \frac{xs}{s-z} \quad y^* = \frac{ys}{s-z}$$

Para el objeto $z < 0$, $s - z > 0$ y $x^* y^*$ son del mismo signo como x, y .

Proyección estereográfica

El aumento de la percepción en la profundidad en una escena 3D es importante en muchas aplicaciones. Hay dos tipos básicos

de percepción de profundidad (usados para el sistema ojos-cerebro): monocular y binocular, dependiendo de la apariencia al usar o no los dos ojos. Las principales señales monocular son:

- **Perspectiva:** convergencia de líneas paralelas.
- **Movimiento paralelo:** cuando la cabeza se mueve lateralmente, los objetos cercanos parecen moverse más rápido que los objetos lejanos en una proyección plana.
- **El tamaño relativo de objetos conocidos.**
- **Área coincidente:** Los objetos que aparecen más cerca y enfrente de otros sin cubrirlos totalmente.
- **Luz alta y sombras.**
- **La atenuación de la atmósfera y la imposibilidad de los ojos para resolverlo en detalles finos a distancia.**
- **Acondicionamiento del foco:** los objetos a diferentes distancias requieren diferentes tensión en los músculos de los ojos para enfocar.

Las principales señales binoculares son:

- **La convergencia angular del eje óptico de los ojos.**
- **Disparidad de la retina:** los diferentes lugares de los objetos proyectados en la retina de los ojos son interpretados como diferencias en distancia.

Las señales monoculares producen solo una débil percepción de la profundidad en 3D. Sin embargo, porque el sistema ojos-cerebro fusionan dos imágenes separadas y distintas que provienen de una sola imagen, la señal binocular produce grandes percepciones de profundidad en 3D. La estereografía intenta reproducir una imagen con características análogas a las reales para una visión binocular verdadera. Hay varias técnicas para generar imágenes estéreo (ver. Capítulo 1: Sistemas de visión).

**ESTA TESIS NO SALE
DE LA BIBLIOTECA**

Todas estas técnicas requieren proyecciones de un objeto dentro de un plano, una por medio del ojo izquierdo y otra por el ojo derecho. La figura 3.7 muestra una proyección de un punto P dentro del eje z desde los centros de proyección en $E_L(-e, 0, d_e)$ y $E_D(e, 0, d_e)$ que corresponden al lado izquierdo y derecho del ojos respectivamente.

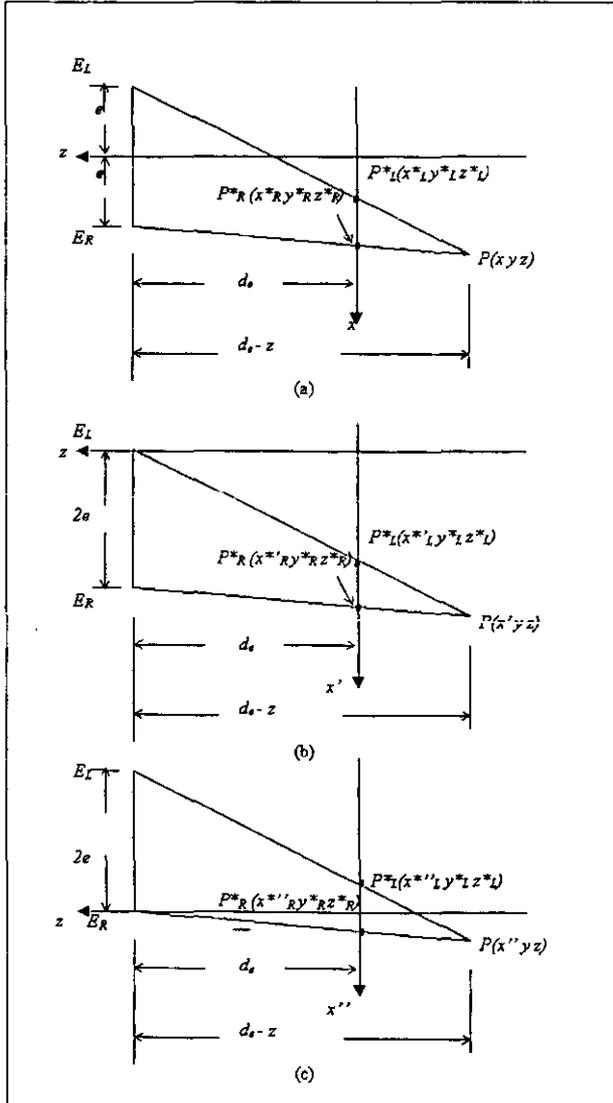


Figura 3.7. Proyección estereográfica.

Por conveniencia, el centro de proyección para el ojo izquierdo es trasladado al eje z como se muestra en la figura 3.7. Usando triángulos semejantes entonces obtenemos:

$$\frac{x_L^*}{d_e} = \frac{x'}{d_e - z}$$

y

$$x_L^* = \frac{x'}{1 - z/d_e} = \frac{x'}{1 + rz}$$

donde $r = -1/d_e$

De manera similar, trasladando el centro de proyección del lado derecho como se muestra en la figura 3.7, y otra vez utilizando triángulos semejantes [14], obtenemos:

$$\frac{x_R^{**}}{d_e} = \frac{x''}{d_e - z}$$

y

$$x_R^{**} = \frac{x''}{1 - z/d_e} = \frac{x''}{1 + rz}$$

$$y^* = \frac{y}{1 - z/d_e} = \frac{y}{1 + rz}$$

La matriz de transformación 4x4 equivalente para el lado izquierdo y derecho son:

$$[S_L] = [T_r, P_r] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ e & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/d_e \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/d_e \\ e & 0 & 0 & 1 \end{bmatrix}$$

y

$$[S_R] = [T_r, P_r] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/d_e \\ -e & 0 & 0 & 1 \end{bmatrix}$$

Consecuentemente, una proyección estereográfica es obtenida por transformación de las escenas usando las ecuaciones anteriores para $[S_L]$ y $[S_R]$.

En visión estereo, la calibración comprende el establecimiento de los parámetros geométricos y de corrección que permiten modelar un ambiente global 3D a partir de sus proyecciones planas conocidas como *par estereo*. Por otro lado, la reconstrucción comprende la generación del ambiente global 3D a partir del modelo y de pares de imágenes. El apareamiento estereo es una herramienta auxiliar de procesamiento de imágenes utilizada en los procesos de calibración y reconstrucción para la correlación de pares de puntos o regiones de las imágenes.

3.2 Calibración

Tiene como objetivo estimar los parámetros de un modelo de cámara definido sobre la base del conocimiento 3D en el sistema de coordenadas global y sus proyecciones en el plano 2D.

El modelo de la figura 3.8 representa un sistema de coordenadas globales X, Y, Z y dos de sus proyecciones planas X^D, Y^D, Z^D y X^I, Y^I, Z^I .

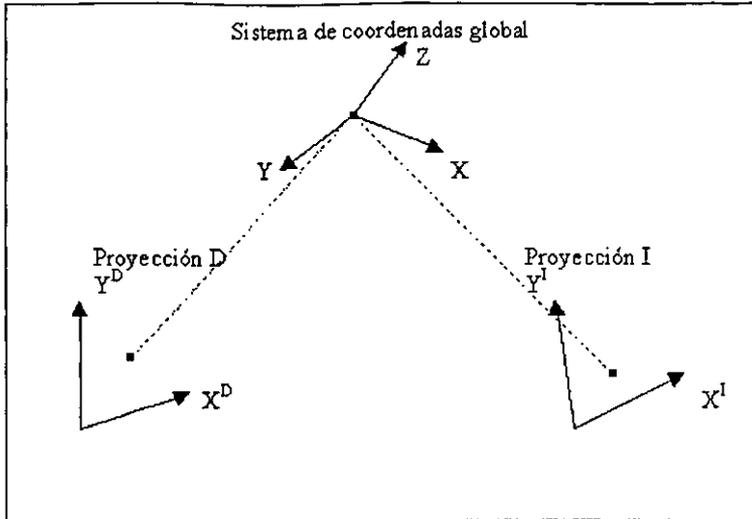


Figura 3.8. Sistema de coordenados usado en el análisis.

El modelo matemático que traslada el sistema de coordenadas globales $[x \ y \ z]$ en una proyección del tipo $[x^* \ y^* \ 0]$ se puede plantear de la siguiente forma:

$$[x, y, z] = [T][x^* \ y^* \ 0] \quad (3.1)$$

preferentemente en coordenadas homogéneas:

$$[x \ y \ z \ 1][T] = [x^* \ y^* \ 0 \ h]$$

$$[x \ y \ z \ 1][T] = h[x^* \ y^* \ 0 \ 1] \quad (3.2)$$

expandiendo la ecuación matricial (3.2) en todos sus términos:

$$[x \ y \ z \ 1] \begin{bmatrix} T_{11} & T_{12} & 0 & T_{14} \\ T_{21} & T_{22} & 0 & T_{24} \\ T_{31} & T_{32} & 0 & T_{34} \\ T_{41} & T_{42} & 0 & T_{44} \end{bmatrix} = h[x^* \ y^* \ 0 \ 1] \quad (3.3)$$

La matriz $[T]$ representa un conjunto de rotaciones, escalamientos y traslaciones sobre el sistema coordenado global $[x \ y \ z \ 1]$ que resultan en la proyección plana $[x^* \ y^* \ 0 \ 1]$. Evidentemente en el modelo de la figura 3.5 existen dos matrices $[T]$ diferentes que representan las proyecciones derecha (D) e izquierda (I). Por simplicidad, en el análisis posterior se considera una sola proyección. Note que x^* y y^* son coordenadas de la proyección en perspectiva sobre el plano $z = 0$, donde $*$ puede ser sustituido por D o I ($[x^D \ y^D \ z^D \ 1]$ y $[x^I \ y^I \ z^I \ 1]$).

La ecuación matricial (3.3) puede expresarse de la siguiente forma:

$$T_{11} x + T_{21} y + T_{31} z + T_{41} = h x^* \quad (3.4a)$$

$$T_{12} x + T_{22} y + T_{32} z + T_{42} = h y^* \quad (3.4b)$$

$$T_{14} x + T_{24} y + T_{34} z + T_{44} = h \quad (3.4c)$$

Despejando h de la ecuación (3.4c) y sustituyendo en (3.4b) y (3.4a) tenemos:

$$(T_{11} - T_{14} x^*) x + (T_{21} - T_{24} x^*) y + (T_{31} - T_{34} x^*) z + (T_{41} - T_{44} x^*) = 0 \quad (3.5a)$$

$$(T_{12} - T_{14} y^*) x + (T_{22} - T_{24} y^*) y + (T_{32} - T_{34} y^*) z + (T_{42} - T_{44} y^*) = 0 \quad (3.5b)$$

Este par de ecuaciones puede ser considerado de tres formas diferentes, las cuales se describen en las siguientes secciones.

Sistema de coordenadas fijo, proyección en perspectiva sobre un plano de proyección perpendicular

Asuma que en las ecuaciones (3.5) x y y y z son conocidos. Entonces tenemos un sistema de dos ecuaciones con dos incógnitas x^* y y^* . De este modo, se puede resolver para encontrar las coordenadas de la proyección plana en perspectiva.

Elementos para determinar la reconstrucción

Como una segunda manera de considerar las ecuaciones (3.5), note que si la localización de ciertos puntos que aparecen en la proyección en perspectiva son conocidos en el sistema de coordenadas global y en la proyección plana en perspectiva, entonces es posible determinar los elementos de la matriz de transformación, es decir los T_{ij} 's. Estos elementos de transformación pueden subsecuentemente ser utilizados para determinar la localización de puntos intermedios utilizando la tercera técnica que se describirá posteriormente en la sección 3.2. para observar como determinar los elementos de reconstrucción, escribamos las ecuaciones (3.5) de la siguiente forma:

$$T_{11}x + T_{21}y + T_{31}z + T_{41} + T_{14}xx' + T_{24}yx' + T_{34}zx' + T_{44}x' = 0 \quad (3.6a)$$

$$T_{12}x + T_{22}y + T_{32}z + T_{42} + T_{14}xy' + T_{24}yy' + T_{34}zy' + T_{44}y' = 0 \quad (3.6b)$$

Asumiendo que x' y y' así como x , y y z son elementos conocidos, las ecuaciones (3.6a) y (3.6b) representan dos ecuaciones con 12 elementos de reconstrucción desconocidos T_{ij} 's. Aplicando estas ecuaciones a 6 puntos conocidos no coplanares en el sistema de coordenadas global y en la proyección plana en perspectiva el sistema se convierte en 12 ecuaciones con 12 incógnitas T_{ij} 's. Entonces la transformación que produce una proyección plana en perspectiva, por ejemplo una fotografía, se puede determinar. Note que en este caso no es necesario conocer previamente la transformación. Si por ejemplo, las proyecciones planas en perspectiva son fotografías, no se requiere la localización ni la orientación de las cámaras [15]. En forma matricial el sistema de doce ecuaciones puede escribirse a partir de las ecuaciones (3.6) de la siguiente forma:

$$\begin{bmatrix} x1 & 0 & -x1x1^* & y1 & 0 & -Y1x1^* & z1 & 0 & -Z1x1^* & 1 & 0 & -x1^* \\ 0 & x1 & -x1y1^* & 0 & y1 & -Y1y1^* & 0 & z1 & -Z1y1^* & 0 & 1 & -y1^* \\ x2 & 0 & -x2x2^* & y2 & 0 & -Y2x2^* & z2 & 0 & -Z2x2^* & 1 & 0 & -x2^* \\ 0 & x2 & -x2y2^* & 0 & y2 & -Y2y2^* & 0 & z2 & -Z2y2^* & 0 & 1 & -y2^* \\ x3 & 0 & -x3x3^* & y3 & 0 & -Y3x3^* & z3 & 0 & -Z3x3^* & 1 & 0 & -x3^* \\ 0 & x3 & -x3y3^* & 0 & y3 & -Y3y3^* & 0 & z3 & -Z3y3^* & 0 & 1 & -y3^* \\ x4 & 0 & -x4x4^* & y4 & 0 & -Y4x4^* & z4 & 0 & -Z4x4^* & 1 & 0 & -x4^* \\ 0 & x4 & -x4y4^* & 0 & y4 & -Y4y4^* & 0 & z4 & -Z4y4^* & 0 & 1 & -y4^* \\ x5 & 0 & -x5x5^* & y5 & 0 & -Y5x5^* & z5 & 0 & -Z5x5^* & 1 & 0 & -x5^* \\ 0 & x5 & -x5y5^* & 0 & y5 & -Y5y5^* & 0 & z5 & -Z5y5^* & 0 & 1 & -y5^* \\ x6 & 0 & -x6x6^* & y6 & 0 & -Y6x6^* & z6 & 0 & -Z6x6^* & 1 & 0 & -x6^* \\ 0 & x6 & -x6y6^* & 0 & y6 & -Y6y6^* & 0 & z6 & -Z6y6^* & 0 & 1 & -y6^* \end{bmatrix} \begin{bmatrix} T_{11} \\ T_{12} \\ T_{14} \\ T_{21} \\ T_{22} \\ T_{24} \\ T_{31} \\ T_{32} \\ T_{34} \\ T_{41} \\ T_{42} \\ T_{44} \end{bmatrix} = 0 \quad (3.7)$$

En donde los subíndices corresponden a los puntos con posiciones conocidas. La ecuación 3.7 es homogénea, ésta contiene un factor de escala arbitrario. Por lo tanto T_{44} puede, por ejemplo, definirse como unitario y el resultado de la transformación normalizarse. Esto reduce los requerimientos a 11 ecuaciones o $5\frac{1}{2}$ puntos. Si la transformación es normalizada, entonces la última columna en $[C']$ se mueve a la derecha y la ecuación matricial no homogénea puede resolverse [ver capítulo 5].

3.3 Reconstrucción

Tiene como objetivo la reconstrucción de coordenadas 3D de un punto P a partir del conocimiento de dos de sus proyecciones 2D derecha (X^D, Y^D) e izquierda (X^I, Y^I) y de los parámetros del modelo estimado como parte de la calibración.

Considerando de una tercer forma las ecuaciones (3.5). Si T , x^* y y^* son conocidos. Entonces tenemos un sistema de dos ecuaciones con tres incógnitas x , y y z . Sin embargo si se dispone de dos proyecciones en perspectivas planas, digamos dos fotografías, entonces las ecuaciones (3.5) pueden escribirse para ambas proyecciones. Esto sería:

$$(T_{11}^D - T_{14}^D x^{*D})x + (T_{21}^D - T_{24}^D x^{*D})y + (T_{31}^D - T_{34}^D x^{*D})z + (T_{41}^D - T_{44}^D x^{*D}) = 0$$

$$(T_{12}^D - T_{14}^D y^{*D})x + (T_{22}^D - T_{24}^D y^{*D})y + (T_{32}^D - T_{34}^D y^{*D})z + (T_{42}^D - T_{44}^D y^{*D}) = 0$$

$$(T_{11}^I - T_{14}^I x^{*I})x + (T_{21}^I - T_{24}^I x^{*I})y + (T_{31}^I - T_{34}^I x^{*I})z + (T_{41}^I - T_{44}^I x^{*I}) = 0$$

$$(T_{12}^I - T_{14}^I y^{*I})x + (T_{22}^I - T_{24}^I y^{*I})y + (T_{32}^I - T_{34}^I y^{*I})z + (T_{42}^I - T_{44}^I y^{*I}) = 0$$

En donde los superíndices D e I identifican el número de proyección plana en perspectiva. Note que las transformaciones $[T^D]$ y $[T^I]$ necesitan ser diferentes. Estas ecuaciones pueden escribirse de nuevo en forma matricial de la siguiente manera:

$$[A][X] = [B] \quad (3.8)$$

En donde:

$$[A] = \begin{bmatrix} T_{11}^D - T_{14}^D * D & T_{21}^D - T_{24}^D * D & T_{31}^D - T_{34}^D * D \\ T_{12}^D - T_{14}^D * D & T_{22}^D - T_{24}^D * D & T_{32}^D - T_{34}^D * D \\ T_{11}^I - T_{14}^I * I & T_{21}^I - T_{24}^I * I & T_{31}^I - T_{34}^I * I \\ T_{12}^I - T_{14}^I * I & T_{22}^I - T_{24}^I * I & T_{32}^I - T_{34}^I * I \end{bmatrix}$$

$$[X]^T = [x \ y \ z]$$

$$[B]^T = - [T_{41}^D - T_{44}^D * D \quad T_{42}^D - T_{44}^D * D \quad T_{41}^I - T_{44}^I * I \quad T_{42}^I - T_{44}^I * I]$$

La ecuación (3.8) representa cuatro ecuaciones en el espacio de tres dimensiones global con incógnitas x , y y z . $[A]$ no es una matriz cuadrada y consecuentemente no puede invertirse para despejar la incógnita $[X]$. Entonces se dice que el problema está sobre-especificado y por lo tanto sólo puede ser resuelto tomando algún promedio o el mejor sentido común.

Una solución promedio puede calcularse recordando que la multiplicación de una matriz por su transpuesta siempre es cuadrada. Por lo tanto, multiplicando ambos lados de la matriz (3.8) por $[A]^T$ resulta

$$[A]^T[A][X] = [A]^T[B]$$

Tomando la inversa de $[[A]^T[A]]$ encontramos una solución para $[X]$

$$[X] = [[A]^T[A]]^{-1}[A]^T[B] \quad (3.9)$$

Si no existe solución para $[X]$, entonces las condiciones impuestas son redundantes y no existe una solución única que proporcione un error menor.

3.4 Apareamiento estéreo

El análisis de movimiento para el apareamiento estéreo es una técnica importante para la correlación de pares de puntos. El movimiento 2D es una proyección del movimiento 3D en escenas reales. Entonces, el modelo de movimiento de 2D describe la relación entre regiones de imágenes sucesivas, proporciona un conjunto reducido de parámetros que permitan encontrar la correspondencia entre pares de imágenes. En nuestro trabajo ensayamos con un modelo simplificado de 4 parámetros orientado a regiones, que resulta tener un buen compromiso entre representatividad y economía. Los parámetros son obtenidos mediante la minimización de estimadores del flujo óptico de máxima probabilidad empleando el método del gradiente.

El problema de la estimación de movimiento vista en capítulos anteriores se puede plantar a través de la figura 3.9

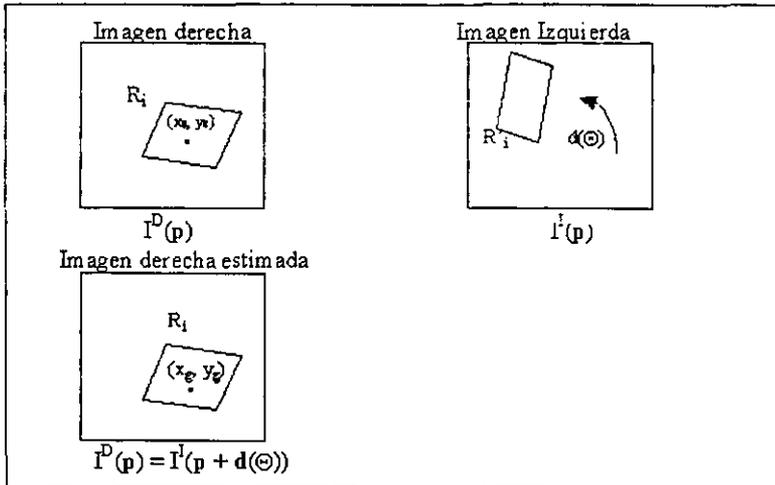


Figura 3.9. Análisis de movimiento.

CAPÍTULO 4

Programa para la estimación de movimiento EMMetro

Tomando en cuenta el estudio anterior, nuestro problema se reduce en encontrar la forma de obtener los valores de la estimación de movimiento. En este sentido, nosotros utilizamos una tarjeta digitalizadora DT3155 para obtener imágenes en tiempo real. Para nuestra interfaz entre la tarjeta y la PC se utilizó Visual Basic 6.0. Nuestro fin es procesar los datos de la tarjeta y enviarlos a la computadora donde serán utilizados para obtener la estimación de movimiento.

En este capítulo proporcionamos información sobre el programa que se desarrolló para una medición sin contacto llamado "Estimación de movimiento, EMMETRO.EXE", describiendo sus principales funciones y características primero a nivel de operador (usuario) y después a nivel programador. Este programa utilizó una tarjeta digitalizadora de imágenes Data Translation DT3155 [17]. Finalmente se enumeran los pasos básicos para analizar una secuencia de imágenes con EMMetro.

4.1 Manual para el operador

4.1.1 Información general

El programa EMMetro es un programa que calcula los cuatro parámetros del modelo lineal basado en el método del gradiente. Utilizando la tarjeta digitalizadora de imágenes DT3155 obtenemos proyecciones de la imagen y, por lo tanto, del par estéreo capturados. El programa EMMetro fue desarrollado bajo un sistema operativo Windows 98 y un lenguaje de programación Visual Basic 6.0, el cual nos proporciona una interfaz amigable para cualquier usuario relacionado con el concepto de estimación de movimiento, figura 4.1.

Los requerimientos básicos para la operación del programa EMMetro son:

- Computadora PC Pentium a 100 MHz, y 32 MB de RAM.
- Tarjeta de digitalización de imágenes Data Translation DT3155.
- Última versión del driver de la tarjeta.
- Dos cámaras CCD de alta resolución Polaris.
- Fuente de poder de 12 Volts.
- Sistema Operativo Windows 95, Windows 98 o NT.
- Instalación de la librería DTAOLFG.DLL en el directorio *windows/system*.

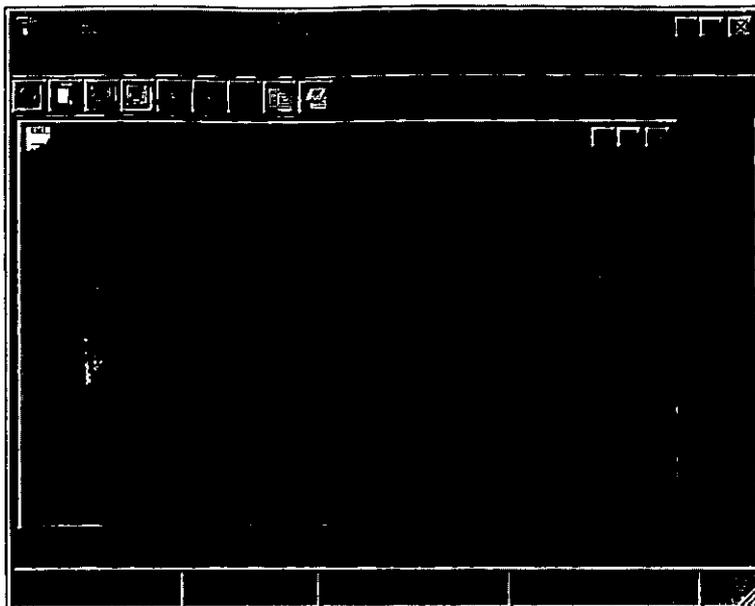


Figura 4.1. Programa “Estimación de movimiento EMMetro”.

El programa EMMetro desempeña las siguientes funciones principales:

- Captura de imágenes en tiempo real.
- Selección de dos fuentes de imagen.
- Extracción de los cuatro parámetros del modelo lineal de la estimación de movimiento.
- Opción de copiar los datos en el Portapapeles.

4.1.2 Descripción de EMMetro

EMMetro cuenta con un menú, que se muestra a continuación en la figura 4.2, y se describe cada una de las funciones:



Figura 4.2. Menú.

Archivo: Manejo de archivos, abre y cierra una sesión con la tarjeta DT3155.

Ver: Barra de herramientas y de estado.

Edición: Opciones del portapapeles.

Imagen: Captura de imágenes y selección de cámaras.

Operaciones: Herramientas de análisis de secuencias de imágenes y el proceso de estimación de movimiento.

Ayuda: Opciones de ayuda.

Barra de herramientas

Esta barra de herramienta, figura 4.3, se encuentra debajo del menú principal y permite el acceso a las funciones más importantes del sistema, que también están establecidas en el menú.

De izquierda a derecha, se describe lo que realiza cada icono de la barra de herramientas.



Figura 4.3. Barra de herramientas.

Funciones de la barra de herramientas:

Nuevo: Crea una sesión nueva y abre la tarjeta DT3155 para capturar imágenes.

Establecer región: Selecciona los límites de la región que se quiere estimar, se muestra en la figura 4.4. Estos valores solo aceptan números impares.

Esto se debe a que nuestra región seleccionada necesita un número entero de píxeles alrededor del centroide; siendo el centroide el centro geométrico de la región.



Figura 4.4. En esta caja de dialogo se establece la región.

Habilitar cámara 1: Habilita la cámara 1 para capturar imágenes.

Habilitar cámara 2: Habilita la cámara 2 para capturar imágenes.

Primera imagen: Captura imagen 1

Segunda imagen: Captura imagen 2.

Proceso de minimización Realiza la operación de estimar el movimiento, y calcula los cuatro parámetros del modelo lineal.

Copiar: Copia los parámetros estimados que proporcionan el mínimo DFD y los pone en el portapapeles.

Portapapeles: Abre el portapapeles.

Barra de estado

En la parte más baja de la ventana de EMMetro, encontramos esta barra que cuenta con varios indicadores: en la primera casilla se muestra el primer DFD que se estimo entre las

dos imágenes capturadas, en las siguientes casillas se registran los valores de las cinco iteraciones conteniendo los cuatro parámetros estimados de movimiento y el DFD, figura 4.5. Al finalizar las iteraciones, la barra de estado despliega los parámetros que proporcionan la mínima DFD, incluyendo la misma DFD mínima.



Figura 4.5. Barra de estado.

Nota: Los cuatro parámetros del algoritmo de gradiente que mencionamos en el capítulo 2 en nuestro programa alfa corresponde a k y lambda corresponde al valor θ .

Menú Archivo

El menú Archivo, figura 4.6, tiene las siguientes funciones:



Figura 4.6. Menú Archivo.

Nuevo Ctrl+N: Crea una nueva sesión para capturar imágenes y habilita el programa controlador de la tarjeta, esta opción tiene la facilidad de poder habilitarse con las teclas Ctrl+N. A continuación se muestra en la figura 4.7.

En la figura 4.7, se muestra la pantalla que indica que la tarjeta DT3155 esta disponible para comenzar a capturar imágenes, en caso contrario mandara un mensaje de error “No Open Device”. Esto podría suceder por no tener conectadas las cámaras o no tener el voltaje adecuado que se les debe suministrar a las cámaras.

Cerrar: Cierra la sesión activa.

Salir: Sale de la aplicación.

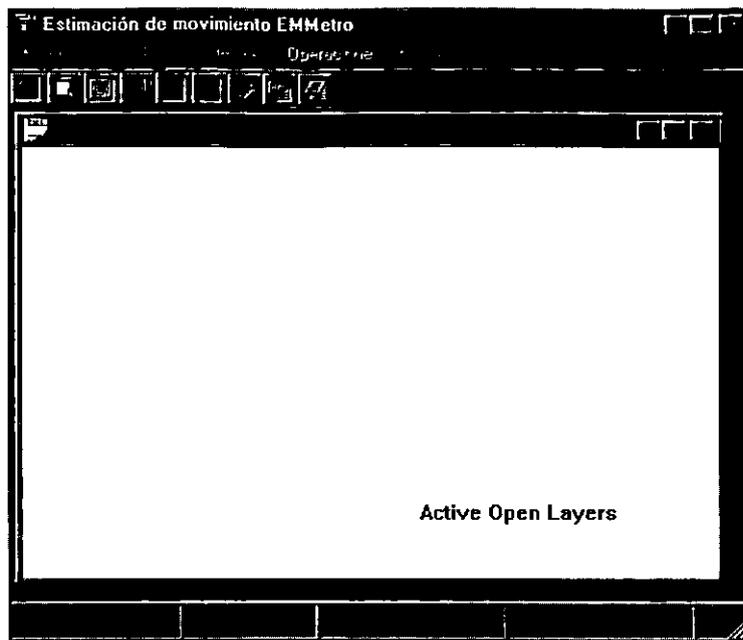


Figura 4.7. Abre una nueva sesión.

Menú Ver

El menú Ver, mostrado en la figura 4.8, tiene la característica de poder habilitar o deshabilitar tanto la barra de herramientas como a la barra de estado para que sean o no visualizadas en la pantalla.

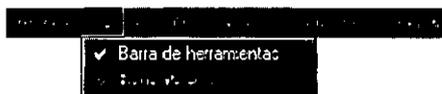


Figura 4.8. Menú Ver.

Menú Edición

En este menú podemos encontrar las siguientes opciones, ver figura 4.9:



Figura 4.9. Menú Edición.

Copiar Xg, Yg: Copia los valores del centroide de la región que nosotros seleccionamos al portapapeles.

Copiar Xg+dx, Yg+dy: Copia el valor del vector estimado de movimiento más las coordenadas del centroide al portapapeles.

Copiar: Copia los valores de los cuatro parámetros que proporcionan la mínima DFD del modelo lineal al portapapeles.

Menú Imagen

En este menú se encuentran funciones que controlan e interactúan con la tarjeta DT3155. El menú se muestra en la figura 4.10.



Figura 4.10. Menú Imagen.

Las opciones de la figura 4.10, ya se mencionaron en la sección de barra de herramientas, a continuación solo mencionaremos la opción de Coordenadas que es la única que no se encuentra en la barra.

Coordenadas: Esta opción nos sirve para volver a mostrar el control que representa los límites de la región que nosotros seleccionamos anteriormente. Esto es con la finalidad de volver

a establecer la región de la cual deseamos estimar su movimiento.

Menú Operaciones



Figura 4.11. Menú Operaciones.

El menú cuenta con las siguientes funciones.

Minimización ctrl+M: Realiza el proceso de minimización. Calcula los cuatro parámetros del modelo lineal de estimación de movimiento.

Dfd: Realiza la Dfd de las dos imágenes capturadas.

Las siguientes opciones son pruebas que se realizaron para la depuración de nuestro programa.

Interpola: Realiza la operación de interpolación de ciertos valores.

Gradiente: Realiza el Gradiente de x de ciertos valores.

Diferencia: Realiza la función de diferencia píxel a píxel entre imágenes.

Menú Ayuda

El menú ayuda muestra una caja de diálogo con información del programa EMMetro.



Figura 4.12. Menú Ayuda.

4.2 Programación Visual Basic

4.2.1. Generalidades de Visual Basic

Visual Basic es un producto con una interfaz gráfica de usuario que sirve para crear aplicaciones para Windows, basado en el lenguaje Basic y en la programación orientada a objetos (OOP).

La palabra “Visual” hace referencia al método que se utiliza para crear la interfaz gráfica de usuario. En lugar de escribir numerosas líneas de código para implementar una interfaz, se utiliza el ratón para arrastrar y colocar los objetos prefabricados al lugar deseado dentro de un formulario.

La palabra “Basic” hace referencia al lenguaje Basic (Beginners All-Purpose Symbolic Instruction Code), un lenguaje utilizado por muchos programadores. Visual Basic ha evolucionado a partir del lenguaje Basic original y ahora contiene centenares de instrucciones, funciones y palabras clave, muchas de las cuales están directamente relacionadas con la interfaz gráfica de Windows.

Visual Basic incluye como características más sobresalientes las siguientes.

Una biblioteca de clases que da soporte a los objetos Windows tales como ventanas, cajas de diálogo, controles (por ejemplo, etiquetas, cajas de texto, botones de pulsación, etc)

Un control que permite utilizar las cajas de diálogo más comúnmente utilizadas (abrir, guardar como, imprimir, color y fuentes).

Un entorno de desarrollo integrado (editor de texto, intérprete, depurador, examinador de objetos, explorador de proyectos, compilador etc.). Visual Basic fue diseñado para ser un intérprete, lo que favorece la creación y la depuración [16].

Asistentes para el desarrollo de aplicaciones: asistente para aplicaciones, asistente para barras de herramientas, asistente para formularios de datos, asistente para empaquetado y distribución, asistente para crear la interfaz pública de controles *ActiveX*, asistente para páginas de propiedades, asistente para objetos de datos, generador de clases, diseñador de complementos y asistente para migración de documentos *ActiveX*.

Galería de objetos incrustados y vinculados (OLE – Object Linking and Embedding). Esto es, software autocontenido en pequeñas y potentes unidades o componentes software para reutilizar cualquier aplicación.

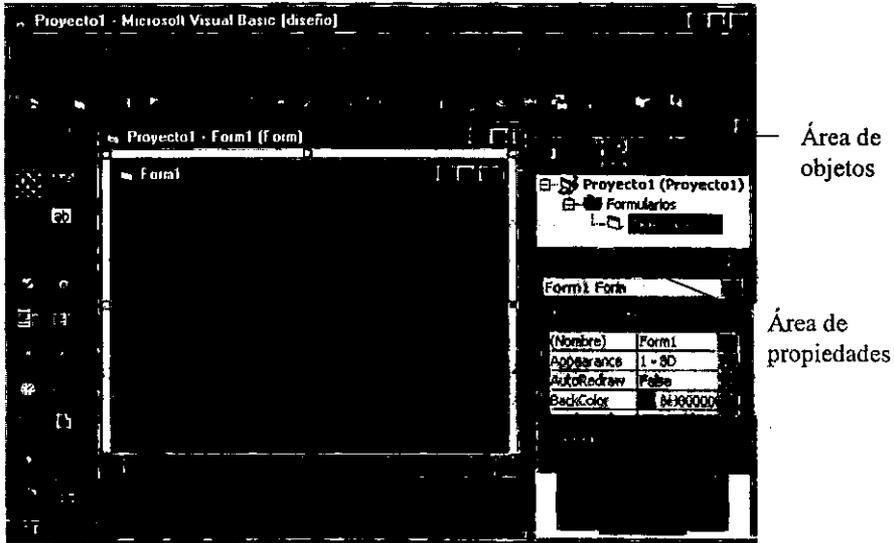
Una interfaz para múltiples documentos (MDI- Multiple Document Interface) que permite crear una aplicación con una ventana principal y múltiples ventanas de documento.

La unidad fundamental de una aplicación Visual Basic es el formulario. Un formulario es realmente una ventana sobre la que se dibujan otros objetos llamados controles, tales como etiquetas, cajas de texto, marcos, casillas de verificación botones de opción, botones de pulsación, etc., con fines de aceptar, ejecutar o visualizar datos.

Finalizando el diseño de la interfaz gráfica (medio de comunicación con el usuario de la aplicación) se escribe el código fuente relacionado con cada objeto (ventanas y controles). Esto quiere decir que cada objeto (ventanas y controles) está ligado a un código que permanece inactivo hasta que se produzca el evento que lo activa. Por ejemplo, podemos programar un botón de órdenes (objeto que se puede pulsar) para que responda a un clic del ratón.

Así que, Windows trabaja estableciendo una comunicación entre los objetos mediante mensajes, los cuales son producidos por eventos. Cuando un mensaje llega a un objeto, éste tiene que procesarlo.

La pantalla de Visual Basic, como se muestra en la figura 4.13. cuenta con 3 áreas principalmente, el área de controles, el área de objetos del proyecto y el área de propiedades de los objetos. En el área central como se puede ver se encuentra el formulario y las pantallas donde se puede escribir el código para que el sistema tenga el comportamiento deseado.



Área de controles

Figura 4.13. Pantalla de Visual Basic.

Para crear una aplicación con Visual Basic, hay tres pasos fundamentales:

- 1) Crear la interfaz gráfica del usuario.
- 2) Establecer las propiedades de los objetos.
- 3) Escribir el código asociado con cada objeto.

Un proyecto de Visual Basic cuenta principalmente con los siguientes módulos y objetos (se menciona esto porque son los mas usados en este trabajo).

Módulos de formulario *

Los módulos de formulario pueden contener texto descriptivo del formulario y sus controles, incluyendo los valores de sus propiedades. También pueden contener declaraciones de formulario de constantes, variables y procedimientos externos, así como procedimientos de evento y procedimientos generales. Un formulario es como la pantalla que se ve en la figura 4.13.

Módulos estándar

Los módulos estándar pueden contener declaraciones públicas o a nivel de módulo de tipos, constantes, variables, procedimientos externos y procedimientos públicos. Estos módulos tampoco tienen una representación gráfica y la forma en que se ven en Visual Basic es a través de la carpeta "módulos" en el área de objetos.

Controles ActiveX

Los controles *ActiveX* son controles opcionales que se pueden agregar al cuadro de herramientas y se pueden usar en formularios. Existen controles *ActiveX* adicionales disponibles en diversas fuentes o se pueden crear los propios utilizando Visual Basic. Estos controles aparecen en la figura 4.13, en el lado izquierdo.

Controles estándar

Los controles estándar los proporciona Visual Basic. Los controles estándar siempre están incluidos en el cuadro de herramientas, al contrario de lo que ocurre con los controles *ActiveX* que se pueden agregar y quitar del cuadro de herramientas. Estos controles son los primeros que aparecen en el costado izquierdo de la figura 4.13.

4.2.2 Control ActiveX DT-Active Open Layers

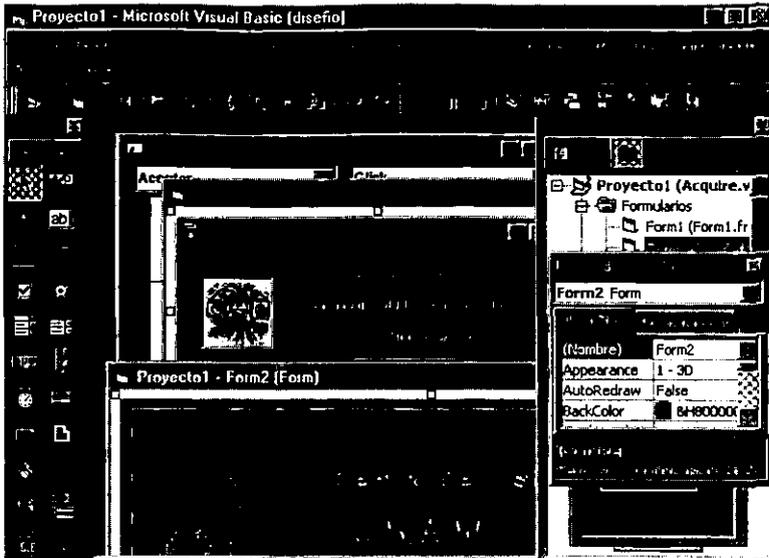
Una vez mencionadas las características más importantes del entorno de programación de Visual Basic, describiremos con más detalle los controles *ActiveX* ya que estos se utilizaron para la programación de la tarjeta DT3155 de Data Translation. Estos controles se llaman DT-Active Open Layers"[17].

Un control *ActiveX* es un objeto que encapsula propiedades, atributos y eventos, y que se ejecuta en el mismo proceso que la aplicación cliente.

Los controles DT-Active Open Layers, específicamente permiten desarrollar aplicaciones de visión en Visual Basic, estos controles proveen las siguientes características.

Conforma un estándar para el DT-Open Layers. Esto quiere decir que es compatible con otras versiones anteriores de Data Translation. No se necesita volver hacer un nuevo código cuando se agregan nuevas tarjetas de Data Translation.

Tiene soporte para múltiples tarjetas. Los controles DT-Active Open Layers son independientes del hardware. Se pueden agregar nuevas tarjetas sin la necesidad de volver a compilar la aplicación.



DT-Active Open Layers

Figura.4.14. Muestra el control DT-Active Open Layers.

A continuación se mencionan algunas propiedades, métodos y eventos que se utilizaron en el programa EMMetro referente al control DT-Active Open Layers.

Generales

Nombre Propiedad/Método	Descripción
OpenDevice método	Abre la tarjeta usando su alias.
CloseDevice método	Cierra la tarjeta previamente abierta con OpenDevice
GetDeviceName método	Regresa el alias para especifica la tarjeta
DeviceCount propiedad	Regresa el numero de tarjetas habilitadas
ActiveDevice propiedad	Activa la tarjeta especificada.

Alojamiento de secuencias

Nombre Propiedad/Método	Descripción
AllocateFrame método	Almacena un bufer para adquirir una imagen y regresa el índice asociado con el bufer.
DestroyFrame método	Destruye un bufer previamente creado por el método AllocateFrame.
FrameCount propiedad.	Establece y regresa el numero de bufers almacenados.
ActiveFrame propiedad	Establece y regresa el índice del cuadro activado.
FrameTop propiedad	Regresa la posición de la primera línea del bufer activo.
FrameLeft propiedad	Regresa la posición del primer pixel del bufer activo.
FrameBaseAddress propiedad	Regresa la dirección del bufer activado.
FrameWidth propiedad	Regresa el numero de pixeles por línea en el bufer activo.
FrameHeight propiedad	Regresa el numero de líneas del bufer activo.
FrameType propiedad	Regresa el tipo de cuadro (o campo) que tu quieras adquirir en el bufer activo.
FrameHorizontalScale propiedad	Regresa el porcentaje que el cuadro ha sido escalado horizontalmente.
FrameVerticalScales propiedad	Regresa el porcentaje que el cuadro ha sido escalado verticalmente.
FrameStorageMode propiedad	Regresa el modo en que los datos del bufer están almacenados.
FrameBytesPerPixel propiedad	Regresa el numero de bytes en cada pixel.

Adquisición y Procesamiento de Imagen

Nombre Propiedad/Método	Descripción
AcquireFrames método	Adquiere uno o mas cuadros.
DrawAcquireFrame método	Dibuja y adquiere un cuadro en la ventana.
ReadFrame método	Lee el píxel de una región rectangular de un cuadro y lo almacena en un arreglo.
WriteFrame método	Escribe un píxel desde un arreglo en una región rectangular sin un bufer.

Procedimiento para habilitar la tarjeta

Para habilitar un dispositivo debemos especificar que tarjeta se esta usando, esto se puede hacer con el método `OpenDevice` o la propiedad `ActiveDevice`. Estos métodos requieren un alias, que es el nombre que le damos a la tarjeta cuando la configuramos. Se necesita un alias diferente para cada tarjeta, esto nos permite acceder a mas de una tarjeta en el mismo sistema.

Ejemplo

```
'abre la tarjeta con el alias que le asignamos
DTAOLFG1.ActiveDevice="MyTestBoard"
```

```
'almacena el alias
'abre la tarjeta
```

```
Alias=DTAOLFG1.ActiveDevice
```

Procedimiento para establecer un cuadro

El cuadro es una porción de área activa de vídeo que queremos digitalizar. El alto del cuadro es la primera línea de vídeo relativa para la área activa de vídeo. El lado izquierdo del cuadro es el primer píxel del vídeo relativo para un área activa de vídeo. El ancho del cuadro es el número de píxeles por línea del vídeo. La altura del cuadro es el número de líneas por campo. La figura 4.15. ilustra estas relaciones.

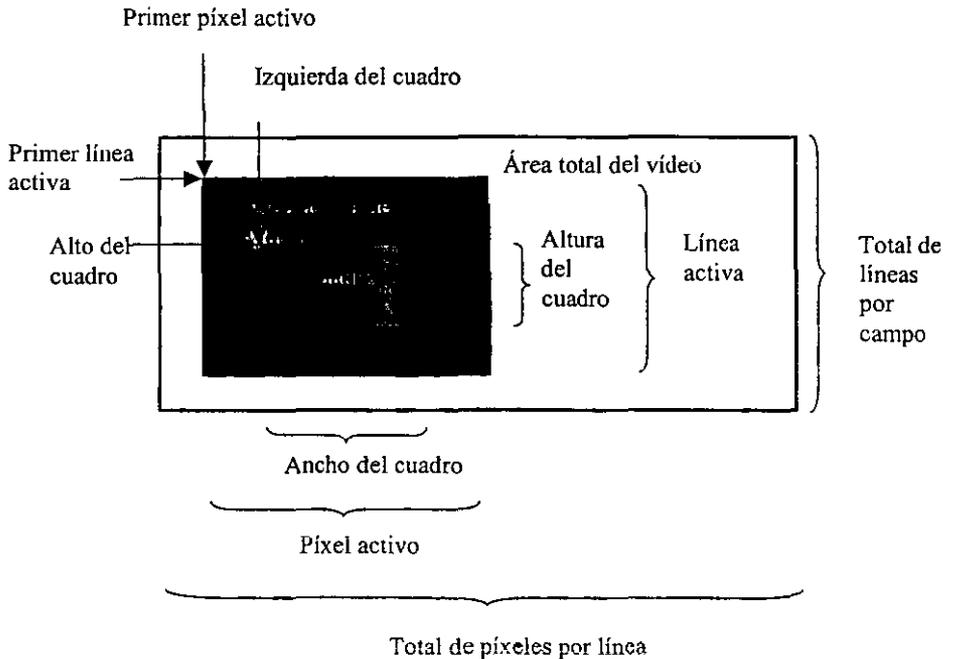


Figura 4.15. Relaciones de Video

Para establecer un cuadro que se quiera digitalizar, antes se debe almacenar un bufer utilizando el método **AllocateFrame**.

Ejemplo para adquirir una o más secuencias.

```
Dim Frameld As Long
```

```
'almacena un buffer
```

```
Frameld = DTAOLFG1.AllocateFrame(0, 0, 640, 480, 100, 100,  
DefaultFrame, DefaultImage)
```

```
'Adquiere la imagen del bufer.
```

```
Call DTAOLFG1.AcquireFrames(1, Frameld, False)
```

Después de adquirir un cuadro, se puede desplegar la imagen en una ventana usando el método **DrawAcquireFrame**, por ejemplo:

'dibuja y adquiere una imagen desde el bufer

Call DTAOLFG1.DrawAcquiredFrame(-1, FrameId)

Operaciones de procesamiento de imágenes

Es importante saber que se pueden extraer los datos de la imagen para poder hacer procesamiento, esto es básico para nuestro programa. A continuación se describen algunas operaciones.

Operación de lectura.- Esto permite mover los datos desde el bufer a un arreglo de usuario.

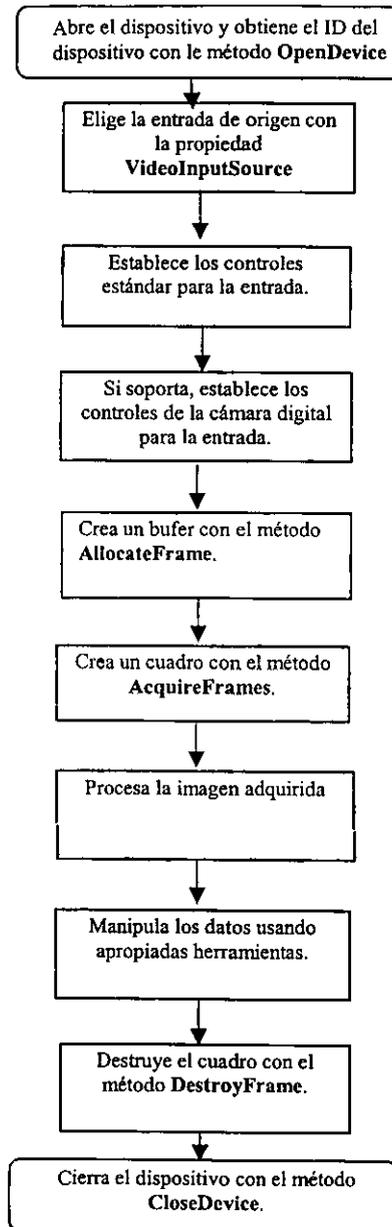
Antes realizar la operación de lectura, es necesario estar seguro que se tiene que crear un arreglo de usuario para almacenar nuestra imagen. Nuestro programa en Visual Basic utiliza un arreglo de usuario bidimensional. El arreglo de usuario debe ser al menos tan grande como el producto de la altura por la anchura del cuadro. Para determinar estos valores fácilmente se pueden utilizar **FrameHeight**, **FrameWidth** and **FrameBytesPerPixel**.

Se usa el método **ReadFrame** para transferir datos desde una región rectangular de un cuadro a un arreglo de usuario. La tarjeta transfiere los datos continuamente.

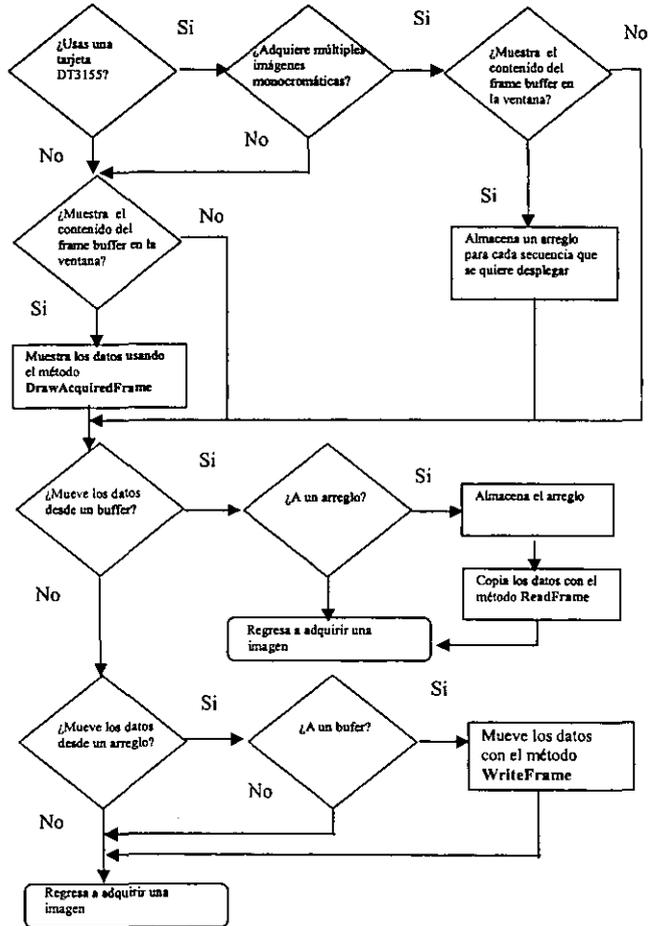
Operación de escritura.- Permite mover los datos desde un arreglo de usuario a un buffer.

Antes de realizar la operación de escritura, se tiene que estar seguro de almacenar un bufer y que el bufer este lo suficientemente grande par poner todos los datos que se quieren transferir [17].

El siguiente es el diagrama de Flujo para adquirir un cuadro.



El siguiente es el diagrama de flujo para adquirir una Imagen.



4.3 Procedimiento típico para analizar secuencias de imágenes

A continuación se describe un procedimiento típico genérico para el análisis de una secuencia de imágenes.

- Encender la PC.
- Polarizar las cámaras con 12 Volts.
- Ajustar los lentes de las cámaras.
- En la PC ejecutar la aplicación EMMETRO.EXE.
- Abrir una nueva sesión.
- Establecer los límites de la región que se quiere estimar.
- Capturar la primera imagen con la cámara 1 ver figura 4.16.

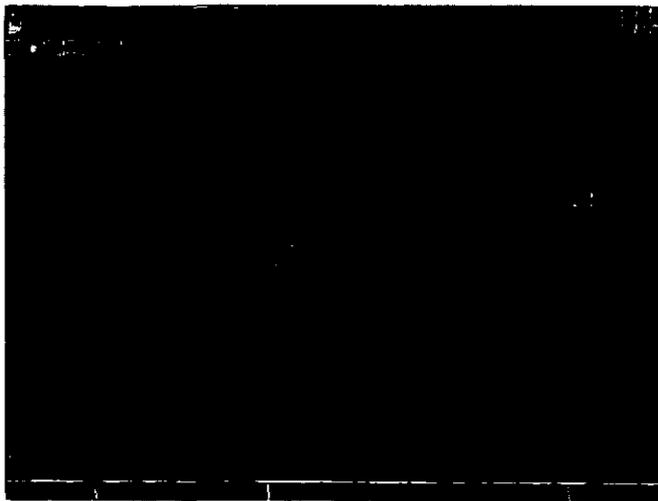


Figura 4.16. Captura imagen 1.

- Arrastrar con el botón derecho o izquierdo del ratón el cuadro de la región que se quiere establecer, ver figura 4.17.

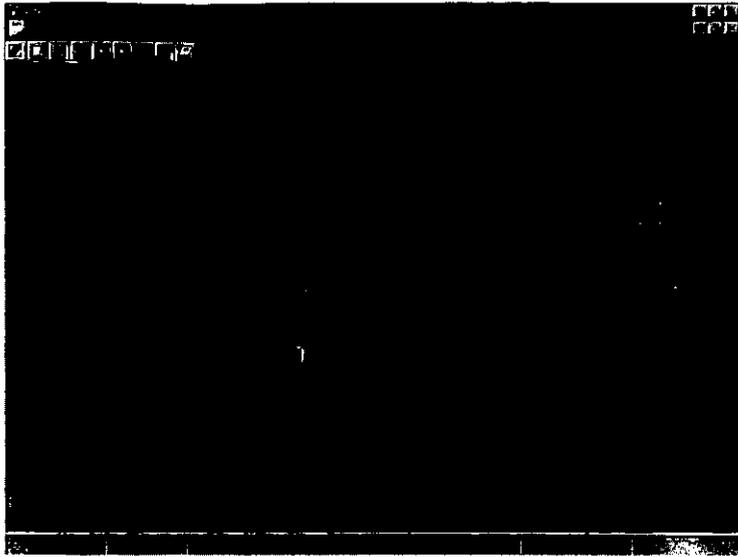


Figura 4.17. Seleccionar región de interés.

- Habilitar la cámara 2 para tomar la segunda imagen.
- Capturar la segunda imagen.
- Ejecutar el procedimiento de minimización, ver figura 4.18.

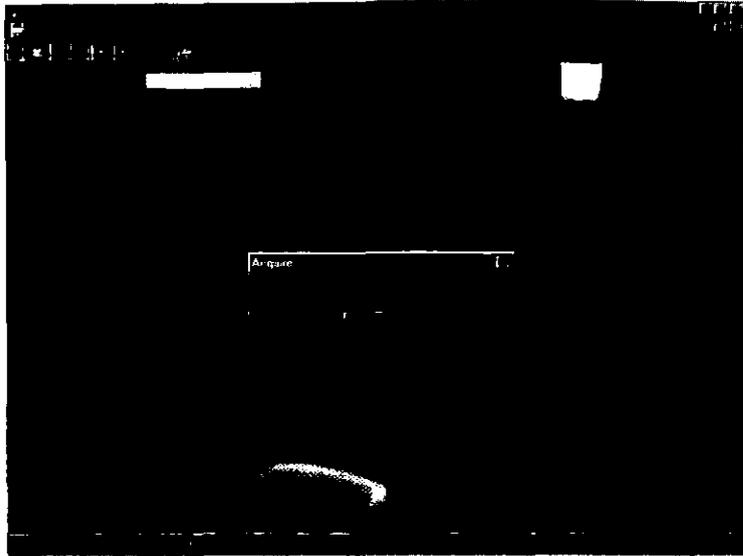


Figura 4.18. Primeros resultados de la DFD natural.

- Copiar los valores estimados, ver 4.19.

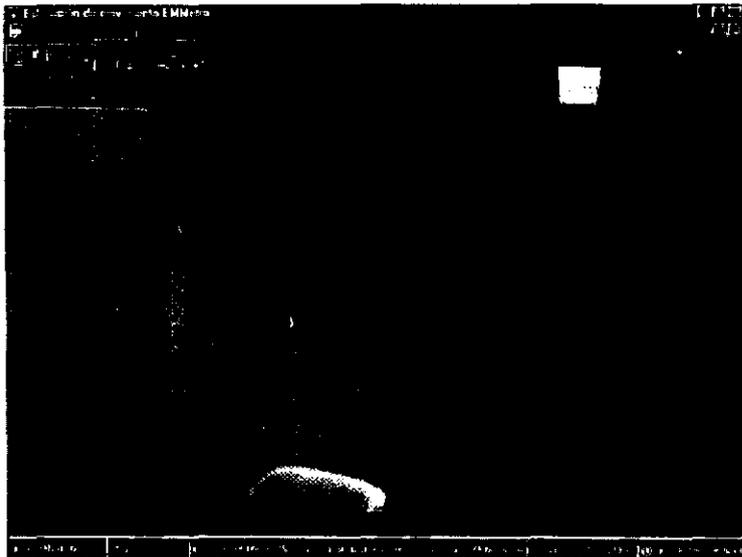


Figura 4.19. Copia automáticamente al portapapeles.

- Abrir el Portapales.

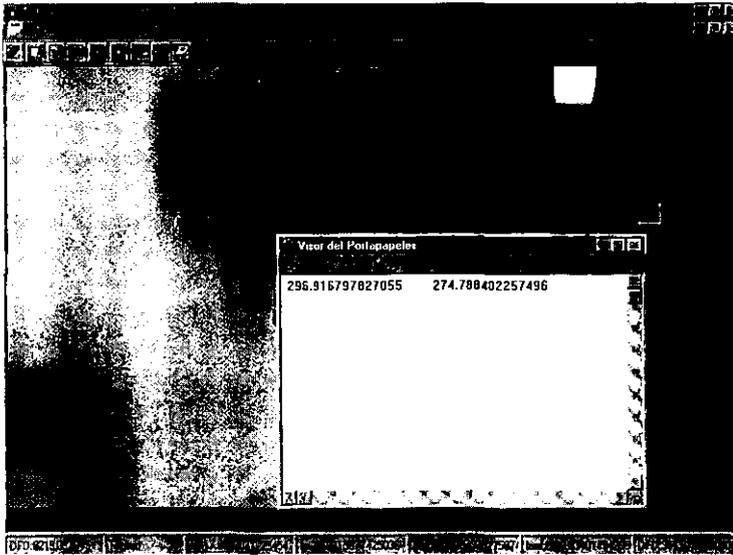


Figura 4.20. Muestra los valores en el portapapeles.

- Transferir los datos del portapapeles como mejor convenga.

CAPÍTULO 5

Resultados y conclusiones

En este capítulo examinaremos los resultados de un experimento obtenidos al operar con el programa “EMMETRO” durante la medición de un objeto utilizando visión estéreo, sin contacto, como una herramienta en la medición y obtención de las coordenadas tridimensionales de un punto o detalle seleccionado. Por lo tanto, analizaremos los resultados y los compararemos con otros procesos de medición. Y al final se darán las conclusiones a las que se llegaron después de haber evaluado el desempeño de la herramienta de análisis de secuencias de imágenes descrita en este trabajo.

5.1 Resultados

En este punto se podrán observar los resultados obtenidos al realizar el proceso de minimización que se explica en este trabajo, pero para poder analizar estos, se tendrán que mencionar los diferentes factores que pueden influir en las mediciones como:

Distorsiones geométricas de la imagen

Podemos considerar tres tipos básicos de distorsión: La primera es causada por la imperfección de la forma de los lentes y se manifiesta solamente por un error de posición radial; considerando que el segundo y tercer tipo de distorsión son generalmente causados por lentes inapropiados y ensambles de cámaras; ambos generan errores radiales y tangenciales en la posición de los puntos, ver figura 5.1.

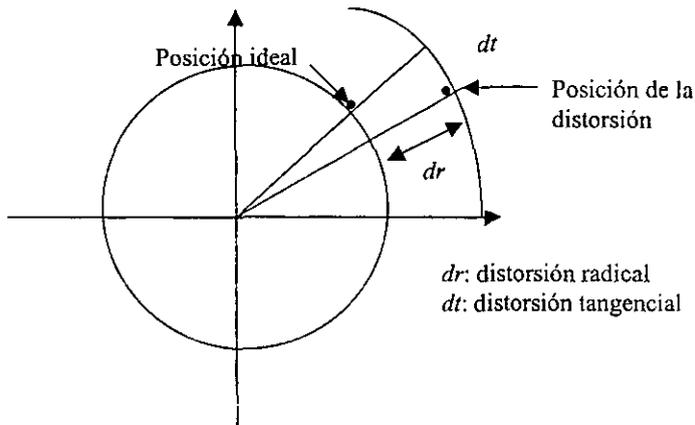


Figura 5.1. Distorsiones radial y tangencial.

- **Distorsión Radial:** La distorsión radial es causada dentro o fuera del espacio del desplazamiento en el cual se da un punto en la imagen desde una locación ideal. Este tipo de distorsión normalmente es causado por la imperfección en la curvatura radial de la curva de los elementos de los lentes. Un desplazamiento radial negativo de los puntos de la imagen es referido como distorsión de "barril". Esto causa que los puntos exteriores se agrupen juntos incrementándose y la escala decrezca. Un desplazamiento positivo radial es referido a la distorsión "pincushion". Esta causa que los puntos exteriores se

esparzan y la escala se incrementa. Este tipo de distorsión es estrictamente simétrica respecto al eje óptico. La figura 5.2. ilustra el efecto de la distorsión radial.

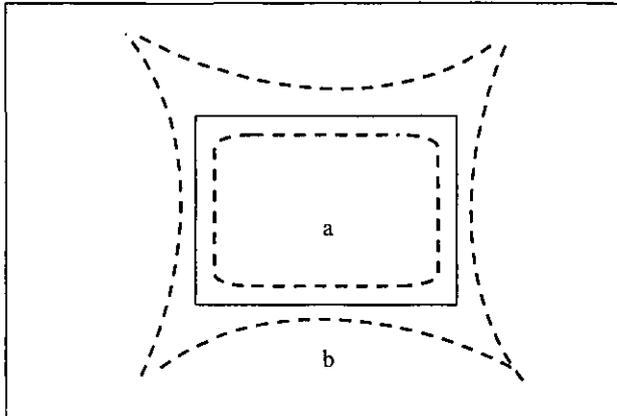


Figura 5.2. Efectos de distorsión radial (a: negativa, b: positiva).

- **Distorsión Descentralizada:** Los sistemas ópticos actuales están sujetos a varios grados de descentralización, es decir los centros ópticos de los elementos de los lentes no son estrictamente colineales. Este defecto introduce lo que es llamado distorsión descentralizada.
- **Distorsión de prisma delgado:** La distorsión de prisma delgado arrastra la imperfección en el diseño y la manufactura de los lentes como el ensamble de la cámara.

La forma utilizada en éste proyecto para minimizar la distorsión de la imagen es mediante el uso de lentes de alta calidad que disminuyen los efectos mencionados.

Resolución espacial

En determinados casos, en los cuales sea indispensable determinar los detalles de la imagen, será necesario un dispositivo con una resolución suficiente para resaltar esos detalles.

La estimación de movimiento

Es necesario determinar el vector de desplazamiento de la región de análisis para tener una reconstrucción confiable de nuestras proyecciones. Por esta razón implementamos un algoritmo de estimación de movimiento (ver Capítulo 2), con esta herramienta nos permite una cooperación entre reconstrucción y apareamiento.

El patrón de calibración

Para nuestro modelo planteado es necesario el uso de una referencia confiable para la calibración de nuestras cámaras. Gracias al Laboratorio de Metrología en el Centro de Instrumentos de la UNAM podemos tener la infraestructura requerida para este propósito.

5.1.1 Simulación en Matlab

Matlab es un ambiente de desarrollo fuertemente orientado a la manipulación numérica. Diseñado por Cleve Moler de la Universidad de Stanford a finales de los años 70, este lenguaje presenta facilidades a ingenieros y físicos equivalentes a las que los diversos intérpretes de comandos (shells) de UNIX ofrecen a los administradores de sistemas y usuarios del sistema. Mathworks, la compañía que lo publica, ha definido Matlab como un sistema interactivo para el cálculo (cómputo) y la visualización, que integra el análisis numérico, el álgebra de matrices, el procesamiento de señales y los gráficos es un ambiente interactivo.

Como lenguaje de programación, Matlab presenta algunas características diferentes a lenguajes más tradicionales (procedurales “duros” como C o Perl). Esto se ha atribuido en gran medida a su línea de evolución: adición de características de gran popularidad entre los usuarios al *corpus* de lenguaje, así como de desarrollos en la implementación de lenguajes de programación universalmente considerados útiles, todo ello con el propósito de ser compatibles con las versiones anteriores del lenguaje.

Hasta recientemente, el único tipo de datos en Matlab era la matriz; un número o escalar, es considerado una matriz de 1×1 , mientras que una cadena es un vector de caracteres. A partir de la versión 5, se añadieron celdas (que extienden la idea de matriz permitiendo ser indicados por cadenas, además de números), estructuras (parecidas a las de C, o los registros de Pascal) y objetos. Estos tipos de datos pueden remediar en alguna medida los defectos en el desarrollo de sistemas mayores mencionados arriba.

Entre las características poco comunes a las que nos referíamos arriba, la más evidente es la marcada preferencia a las operaciones sobre matrices elemento por elemento.

El parecido entre Matlab y los lenguajes de “shell” y orientados al procesamiento de texto, como Perl también tiene sus aspectos negativos: el uso para el que están diseñados es programas rápidos y cortos, con poca o ninguna comunicación con un sistema mayor, lo que no facilita el desarrollo de ese tipo de sistemas. Evidencia de ello es el manejo de variables globales, lo suficientemente aparatoso para descalificarlas como la solución rápida que representan en lenguajes tradicionales. En este sentido, Matlab obliga a mantener un diseño poco acoplado, lo que facilita el mantenimiento del programa, pero no su desarrollo.

Los programadores de Matlab pronto admitieron que había muchas cosas que otros lenguajes hacían mejor que Matlab, por lo que prefirieron proveer un escape tanto al sistema operativo como una interfaz de programación en C, que agregar características al lenguaje original.

Antes de comenzar a programar nuestra interface final, que fue hecha en Visual Basic, se hizo una simulación en Matlab con varias restricciones:

- Se eligieron dos imágenes estáticas, ver figura 5.3.
- La región seleccionada no cambia, siempre es la misma.

Todo esto se hizo para comprobar que nuestro algoritmo de minimización basada en el gradiente del modelo 2D de 4 parámetros opera correctamente.



Figura 5.3. Imagen (a) es *claire10.png* e imagen (b) es *claire11.png*.

Nosotros examinamos una región, esta región es la que corresponde al contorno de la boca, donde en la imagen *a* y *b* se nota como la boca se desplaza ligeramente hacia arriba.

Primero inicializamos nuestras variables a cero en la siguiente tabla se muestran los resultados.

	T_x	T_y	Alfa	Lambda	DFD
Inicial	0	0	0	0	1090.006

En la siguiente etapa realizamos un block matching de la región para obtener los valores iniciales de entrada al proceso de minimización de la DFD.

	Tx	Ty	Alfa	Lambda	DFD
Buscado	0	3	0	0	148.446

En la siguiente tabla se muestran los resultados obtenidos de las cinco iteraciones que realiza el algoritmo de estimación de movimiento (minimización de la DFD).

Iteración	Tx	Ty	Alfa	Lambda	DFD
1.-	0.010	2.898	-0.006	-0.002	118.204
2.-	0.029	2.811	-0.013	-0.003	99.531
3.-	0.053	2.741	-0.018	-0.003	90.908
4.-	0.070	2.698	-0.024	-0.002	89.591
5.-	0.083	2.658	-0.030	-0.001	91.348

Los resultados finales fueron:

	Tx	Ty	Alfa	Lambda	DFD
Mínima DFD	0.070	2.698	-0.024	-0.002	89.591

Como nos podemos dar cuenta, el mejor resultado es el que minimiza la DFD, el algoritmo del gradiente de cuatro parámetros nos garantiza que en 5 iteraciones obtendremos un resultado óptimo.

5.1.2 Estimación de movimiento usando el programa EMMetro

Para realizar el proceso de estimación de movimiento usando el programa EMMetro, en el laboratorio de Metrología del Centro de Instrumentos, se utilizó el siguiente equipo.

- 1) Una computadora PC (Pentium III) a 450 Mhz.

- 2) Dos cámaras CCD de alta resolución Polaris ver figura 5.4.



Figura 5.4. Equipo utilizado.

- 3) Una máquina de medición de coordenadas Zeiss figura 5.5.

A continuación se describe como se realizó el emparejamiento estéreo de regiones de interés usando el programa EMMETRO.



Figura 5.5. Máquina Zeiss..

Primero se definió un volumen de visión, en nuestro caso utilizamos un cubo. Llamamos volumen de visión al espacio necesario para envolver el objeto que se quiera medir, el cual debe ser envuelto por el área de visión de las dos cámaras, ver figura 5.6. Además, las dos cámaras deben de estar ajustadas aproximadamente con la misma intensidad de luz e igualmente enfocadas. Como nos podemos dar cuenta, parece que solo hay una cámara pero no es así; son dos cámaras tomando el mismo objeto.

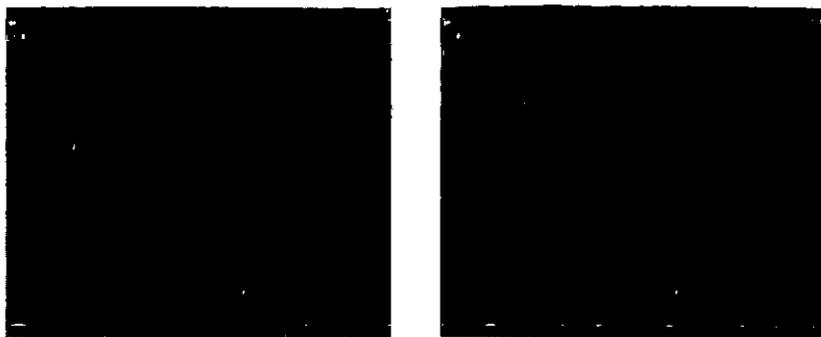


Figura 5.6. Cámara uno y cámara dos.

El siguiente paso fue proceder a tomar seis puntos no coplanares dentro del volumen (el cubo) con el programa EMMetro y además obtener esos mismos puntos con la máquina de coordenadas Zeiss este paso se ilustra en las figuras 5.7 y 5.8 siguientes:

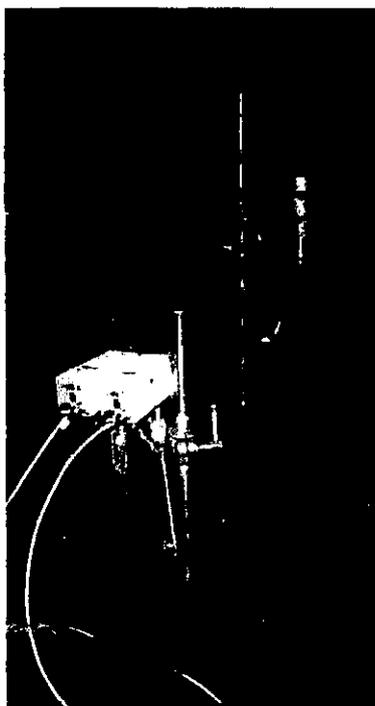


Figura 5.7. Medición de puntos.

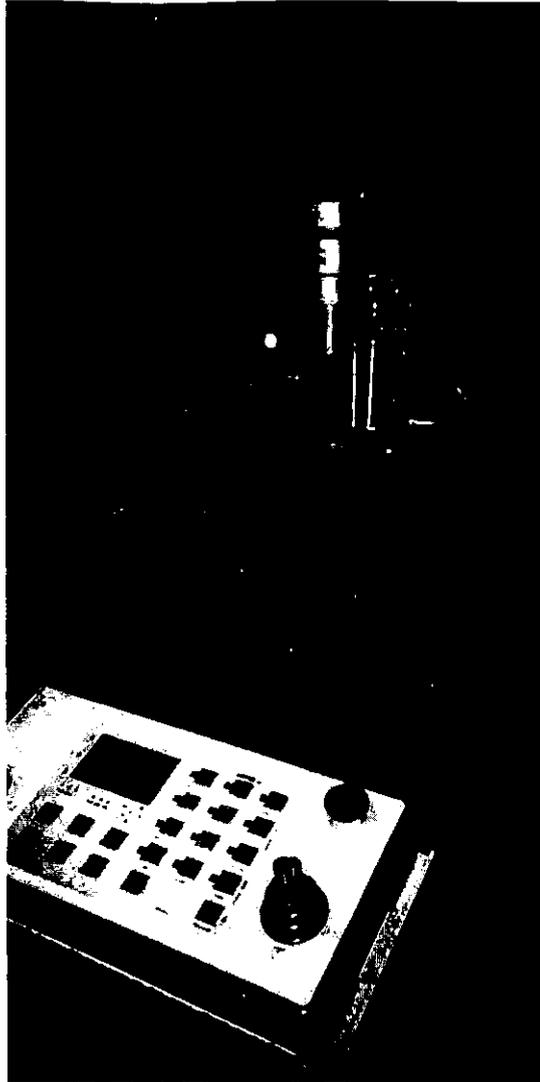


Figura 5.8. Obtención de los seis puntos no coplanares mediante la máquina de coordenadas.

Para la obtención de los seis puntos bidimensionales de nuestro programa EMMetro, se tomaron en cuenta varias restricciones para obtener mejores resultados, estas las mencionamos a continuación:

- Tener un fondo aproximadamente contrastante con la región de interés ya que nuestro algoritmo puede confundir la región seleccionada por otra que no sea de interés, ver figura 5.9.



Figura 5.9. Un fondo contrastante.

Como nos podemos dar cuenta, el fondo de esta imagen es contrastante, no hay objetos en donde la región se pueda confundir.

- Otra restricción en algunos casos fue necesario tapar los otros sensores del palpador, ya que como podemos observar en la figura 5.10, el palpador consta de tres sensores. Para evitar este caso recomendamos utilizar un palpador con un solo sensor. En alguna ocasión nuestro algoritmo nos daba las coordenadas de otro sensor que no deseábamos.

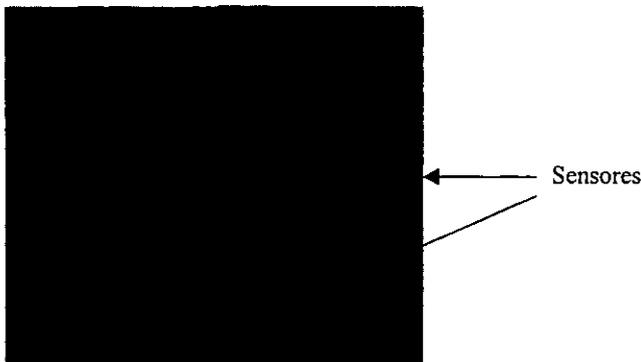


Figura 5.10. Palpador de 5 sensores.

Las coordenadas obtenidas de los seis puntos no coplanares se muestra en la figura 5.11 y en la tabla 5.1, mediante la máquina Zeiss y el proceso de estimación estéreo.

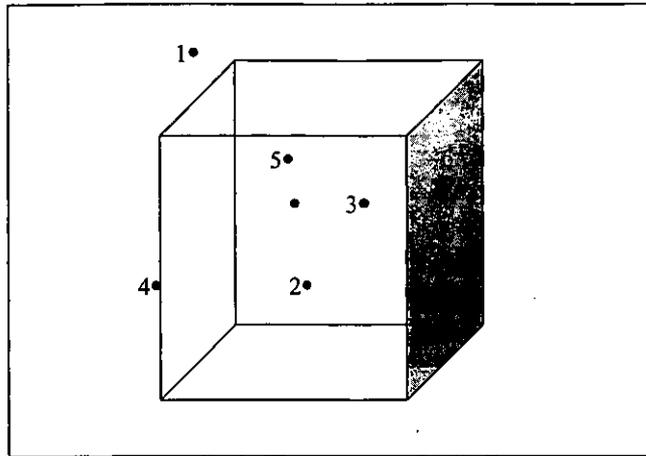


Figura 5.11. Seis puntos no coplanares.

La siguiente tabla, muestra el desempeño del programa MMETRO en el emparejamiento estéreo.

Puntos	Máquina de Coordenadas Zeiss-EMMetro	Programa EMMetro Cámara 1	Programa EMMetro Cámara 2
Uno	(0,0,0)	(493.75)	(465.142,106.494)
Dos	(51.784,5.741,-114.412)	(336,374)	(313.081,416.086)
Tres	(105.985,27.515,-49.656)	(185,195)	(158.154,235.327)
Cuatro	(23.576,-121.447,-78.644)	(474.281)	(416.447.364.713)
Cinco	(91.321,-104.058,9.656)	(324,55)	(263.83,101.770)
Seis	(150.373,-90.625,-24.805)	(180,126)	(120.310,176.164)

Tabla 5.1. Coordenadas de seis punto no coplanares.

5.1.3 Visión estéreo usando el programa EMMetro

Siguiendo con el análisis de los resultados, en este apartado analizaremos las conclusiones en la etapa final del experimento. Con ayuda de la ecuaciones presentadas en el capítulo 3 (Visión estéreo) podremos llegar a obtener la calibración de nuestras cámaras y la reconstrucción de las proyecciones.

Como ya vimos anteriormente, para la calibración necesitamos 6 puntos conocidos no coplanares de nuestro sistema global que obtuvimos con la ayuda de la máquina de coordenadas que se encuentra en el Laboratorio de Metrología. Estos datos nos servirán como referencia en nuestro proceso de calibración y como estimador del error durante la reconstrucción.

De la ecuación 3-7 se obtiene:

$$C = \begin{bmatrix} x1 & 0 & -x1x1^* & y1 & 0 & -Y1x1^* & z1 & 0 & -Z1x1^* & 1 & 0 & -x1^* \\ 0 & x1 & -x1y1^* & 0 & y1 & -Y1y1^* & 0 & z1 & -Z1y1^* & 0 & 1 & -y1^* \\ x2 & 0 & -x2x2^* & y2 & 0 & -Y2x2^* & z2 & 0 & -Z2x2^* & 1 & 0 & -x2^* \\ 0 & x2 & -x2y2^* & 0 & y2 & -Y2y2^* & 0 & z2 & -Z2y2^* & 0 & 1 & -y2^* \\ x3 & 0 & -x3x3^* & y3 & 0 & -Y3x3^* & z3 & 0 & -Z3x3^* & 1 & 0 & -x3^* \\ 0 & x3 & -x3y3^* & 0 & y3 & -Y3y3^* & 0 & z3 & -Z3y3^* & 0 & 1 & -y3^* \\ x4 & 0 & -x4x4^* & y4 & 0 & -Y4x4^* & z4 & 0 & -Z4x4^* & 1 & 0 & -x4^* \\ 0 & x4 & -x4y4^* & 0 & y4 & -Y4y4^* & 0 & z4 & -Z4y4^* & 0 & 1 & -y4^* \\ x5 & 0 & -x5x5^* & y5 & 0 & -Y5x5^* & z5 & 0 & -Z5x5^* & 1 & 0 & -x5^* \\ 0 & x5 & -x5y5^* & 0 & y5 & -Y5y5^* & 0 & z5 & -Z5y5^* & 0 & 1 & -y5^* \\ x6 & 0 & -x6x6^* & y6 & 0 & -Y6x6^* & z6 & 0 & -Z6x6^* & 1 & 0 & -x6^* \\ 0 & x6 & -x6y6^* & 0 & y6 & -Y6y6^* & 0 & z6 & -Z6y6^* & 0 & 1 & -y6^* \end{bmatrix}$$

Como mencionamos en el capítulo 3, podemos definir a T_{44} como unitario y por lo tanto se reduce a una matriz C' de 11×11

$$C' = \begin{bmatrix} x1 & 0 & -x1x1^* & y1 & 0 & -Y1x1^* & z1 & 0 & -Z1x1^* & 1 & 0 \\ 0 & x1 & -x1y1^* & 0 & y1 & -Y1y1^* & 0 & z1 & -Z1y1^* & 0 & 1 \\ x2 & 0 & -x2x2^* & y2 & 0 & -Y2x2^* & z2 & 0 & -Z2x2^* & 1 & 0 \\ 0 & x2 & -x2y2^* & 0 & y2 & -Y2y2^* & 0 & z2 & -Z2y2^* & 0 & 1 \\ x3 & 0 & -x3x3^* & y3 & 0 & -Y3x3^* & z3 & 0 & -Z3x3^* & 1 & 0 \\ 0 & x3 & -x3y3^* & 0 & y3 & -Y3y3^* & 0 & z3 & -Z3y3^* & 0 & 1 \\ x4 & 0 & -x4x4^* & y4 & 0 & -Y4x4^* & z4 & 0 & -Z4x4^* & 1 & 0 \\ 0 & x4 & -x4y4^* & 0 & y4 & -Y4y4^* & 0 & z4 & -Z4y4^* & 0 & 1 \\ x5 & 0 & -x5x5^* & y5 & 0 & -Y5x5^* & z5 & 0 & -Z5x5^* & 1 & 0 \\ 0 & x5 & -x5y5^* & 0 & y5 & -Y5y5^* & 0 & z5 & -Z5y5^* & 0 & 1 \\ x6 & 0 & -x6x6^* & y6 & 0 & -Y6x6^* & z6 & 0 & -Z6x6^* & 1 & 0 \end{bmatrix}$$

de donde

$$\begin{aligned} [C][T] &= [D] \\ [C']^{-1}[C][T] &= [C']^{-1}[D] \\ [T] &= [C']^{-1}[D] \end{aligned} \quad (5.1)$$

y

$$D^T = [x1' \ y2' \ x1' \ y2' \ x3' \ y3' \ x4' \ y4' \ x5' \ y5' \ x6']$$

donde

T es el vector con los elementos de la matriz de translación.

x_n, y_n son las coordenadas globales obtenidas con la máquina de coordenadas.

x_n^*, y_n^* son las coordenadas relativas obtenidas con la ayuda de nuestro programa EMMetro.

Con el apoyo de los datos recabados por el programa EMMetro, para cada proyección y los puntos no coplanares se tiene la tabla 5.2.

Sistema Global			Sistema Proyección		Sistema Proyección 2	
x	y	z	x	y	x	Y
0	0	0	493	75	465.142	106.494
51.784	5.741	-114.412	336	374	313.081	416.086
105.985	27.515	-49.656	185	195	158.154	235.827
23.576	-121.447	-78.644	474	261	416.447	364.713
91.321	-104.058	9.656	324	55	263.83	101.77
150.373	-90.625	-24.805	180	126	120.31	176.164

(a)

(b)

(c)

Tabla 5.2. Datos obtenidos

De la tabla anterior se forman las matrices C'_D y C'_I que corresponden a la proyección derecha e izquierda respectivamente.

$$C_D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 51.784 & 0 & -17399.424 & 5.741 & 0 & -1928.976 & -114.412 & 0 & 38442.432 & 1 & 0 \\ 0 & 51.784 & -19367.216 & 0 & 5.741 & -2147.134 & 0 & -114.412 & 42790.088 & 0 & 1 \\ 105.985 & 0 & -19607.225 & 27.515 & 0 & -5090.275 & -49.656 & 0 & 9186.36 & 1 & 0 \\ 0 & 105.985 & -20667.075 & 0 & 27.515 & -5365.425 & 0 & -49.656 & 9682.92 & 0 & 1 \\ 23.576 & 0 & -11175.024 & -121.447 & 0 & 57565.878 & -78.644 & 0 & 37277.256 & 1 & 0 \\ 0 & 23.576 & -6153.336 & 0 & -121.447 & 31697.667 & 0 & -78.644 & 20526.084 & 0 & 1 \\ 91.321 & 0 & -29588.004 & -104.058 & 0 & 33714.792 & 9.656 & 0 & -3128.544 & 1 & 0 \\ 0 & 91.321 & -5022.655 & 0 & -104.058 & 5723.19 & 0 & 9.656 & -531.08 & 0 & 1 \\ 150.373 & 0 & -27067.14 & -90.625 & 0 & 16312.5 & -24.805 & 0 & 4464.9 & 1 & 0 \end{bmatrix}$$

y

$$C_I = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 51.784 & 0 & -16212.5 & 5.741 & 0 & -1797.3 & -114.412 & 0 & 35820.2 & 1 & 0 \\ 0 & 51.784 & -21546.5 & 0 & 5.741 & -2388.7 & 0 & -114.412 & 47605.2 & 0 & 1 \\ 105.985 & 0 & -16751.9 & 27.515 & 0 & -4351.6 & -49.656 & 0 & 7853.2 & 1 & 0 \\ 0 & 105.985 & -24994.1 & 0 & 27.515 & -6488.7 & 0 & -49.656 & 11710.2 & 0 & 1 \\ 23.576 & 0 & -9818.1 & -121.447 & 0 & 50576.2 & -78.644 & 0 & 32751.0 & 1 & 0 \\ 0 & 23.576 & -7183.9 & 0 & -121.447 & 37006.4 & 0 & -78.644 & 23963.8 & 0 & 1 \\ 91.321 & 0 & -24093.2 & -104.058 & 0 & 27453.6 & 9.656 & 0 & -2547.5 & 1 & 0 \\ 0 & 91.321 & -9293.7 & 0 & -104.058 & 10589.9 & 0 & 9.656 & -982.6 & 0 & 1 \\ 150.373 & 0 & -18091.3 & -90.625 & 0 & 10903.0 & -24.805 & 0 & 2984.2 & 1 & 0 \end{bmatrix}$$

De la ecuación 5.1 y las matrices C'_D y C'_I se obtienen T_D y T_I ya en forma matricial.

$$T_D = \begin{bmatrix} -2.551844343 & -0.0844726368 & 0 & 0.000288899 \\ -1.183327443 & -0.228932749 & 0 & -0.001539495 \\ 0.149263015 & -2.672774061 & 0 & 2.7904E-05 \\ 493 & 75 & 0 & 1 \end{bmatrix}$$

$$T_r = \begin{bmatrix} -2.66650294 & -0.01053472 & 0 & 0.0001644 \\ -0.88320471 & -0.40162878 & 0 & -0.00169644 \\ 0.10852392 & -2.69011113 & 0 & 8.7221E-05 \\ 465.142 & 106.494 & 0 & 1 \end{bmatrix}$$

Como se observo en la sección 3.3 (Reconstrucción) teniendo T y las proyecciones x^* y y^* se obtiene la matriz X que contiene las coordenadas del sistema global. Para nuestro análisis se tienen las siguientes proyecciones $I^D(324, 182)$ y $I^I(280.193, 229.626)$ Por lo tanto se generan la matrices A y B como sigue:

$$A = \begin{bmatrix} -2.645447608 & -0.68453109 & 0.140222122 \\ -0.13461546 & 0.057413305 & -2.677964203 \\ -2.71256694 & -0.407872702 & 0.084085147 \\ -0.048285454 & -0.012080906 & -2.710139383 \end{bmatrix}$$

Y

$$B^T = [-169 \quad 111 \quad -184.949 \quad 123.132]$$

De la ecuación 3-9 se tiene:

$$X = [74.23139155 \quad -49.61087846 \quad -46.44449252]$$

Comparando este resultado con nuestro patrón (la maquina de coordenadas) $X_{real}=(74.231,-49.610, -46.445)$ se tiene que el error promedio relativo es de 1.767146×10^{-3} .

En la figura 5.12, puede apreciarse el proceso de calibración y reconstrucción con la ayuda del la máquina de coordenadas.

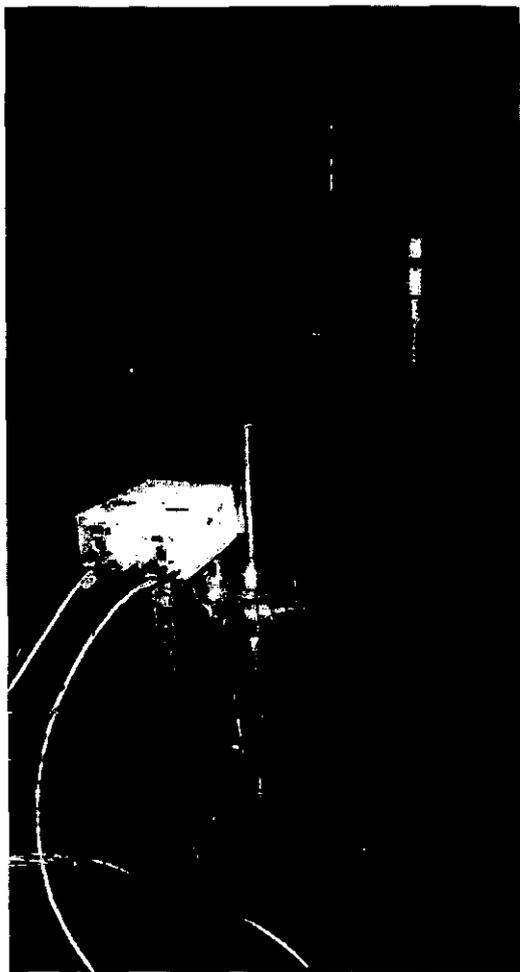


Figura 5.12. Captura de los 6 puntos no coplanares.

5.2 Conclusiones y trabajo a futuro

La conclusión más importante es la función de esta herramienta que servirá como una tecnología alternativa en el departamento de Metrología del Centro de Instrumentos y que a su vez dará servicio a la industria.

Con base en nuestro trabajo podemos mencionar algunas conclusiones que enlistamos a continuación:

Como se ha venido mencionando durante el desarrollo de este trabajo, el bajo costo que implica esta técnica es una de las alternativas de interés para su implementación. Ya que se puede sustituir, en algunos casos, la máquina de coordenadas por el sistema de medición por visión.

La automatización de el análisis estéreo puede facilitarse puesto que el emparejamiento estéreo es realizado por el programa. También ayuda a reducir los errores de apreciación, por lo que la tecnología expuesta en este trabajo ayuda a obtener datos fidedignos. Sin olvidar mencionar que la velocidad de análisis puede ser considerablemente aminorada.

Sin embargo, existen aspectos que pueden resolverse y pueden servir como temas de futuras investigaciones. Haciendo mención, por citar algunos tenemos:

La estimación de movimiento: podrían proponerse métodos no lineales para la estimación de movimiento que consideren problemas como la variación de la luminancia.

Procesos de calibración no-lineal: estas técnicas nos permiten corregir errores ópticos y posibles fallas mecánicas, así como también errores de digitalización.

Finalmente, es agradable participar en un proyecto actual, que ve hacia el futuro, ya que el mundo cada vez requiere más y mejores técnicas de visión, debido a la total automatización del mundo que nos rodea.

ANEXO A

Interpolación

Para reconstruir el movimiento tenemos que conocer cada punto de la imagen t_2 y reconstruir el antecedente de la imagen t_1 , esto lo hacemos teniendo el desplazamiento estimado entre estos dos puntos (calcularemos el desplazamiento de la imagen t_2 hacia la imagen t_1 , por lo tanto en el sentido es inverso al desplazamiento físico).

Sin embargo para reconstruir un punto $p(x, y, t_2)$ conociendo el desplazamiento $\mathbf{d} = (dx, dy)^t$ es necesario conocer la luminancia del "punto" $p(p + dx, y + dy, t_1)$ correspondiente. En algunos casos las coordenadas del vector de desplazamiento $\mathbf{d} = (dx, dy)^t$ no son valores enteros, entonces es necesario determinar la luminancia de un punto reconstruido $\hat{p} = (x, y, t_2) = p(x + dx, y + dy, t_1)$ para interpolar el valor de luminancia.

El tipo de interpolación usado en todo este estudio es bilineal ver figura A.1.

Esta técnica supone que la luminancia entre cuatro píxeles forman un cuadrado elemental una función lineal lo que no es en el caso de los contornos espaciales de la imagen donde se tiene una ruptura brutal de luminancia. Esta interpolación introduce por consiguiente una inclinación en el proceso de evaluación de

movimiento y reconstrucción (principalmente por un contorno espacial).

Matemáticamente, la reconstrucción se hace en cada punto usando la ecuación siguiente:

$$\begin{aligned} p(x, y, t_2) &= p(x + dx, y + dy, t_1) \\ &= ENT(\Delta x \Delta y I_1 + (1 - \Delta x) \Delta y I_2 + dx (1 - \Delta y) I_3 + (1 - \Delta x) (1 - \Delta y) I_4) \end{aligned}$$

donde $ENT(x)$ es la parte entera de x .

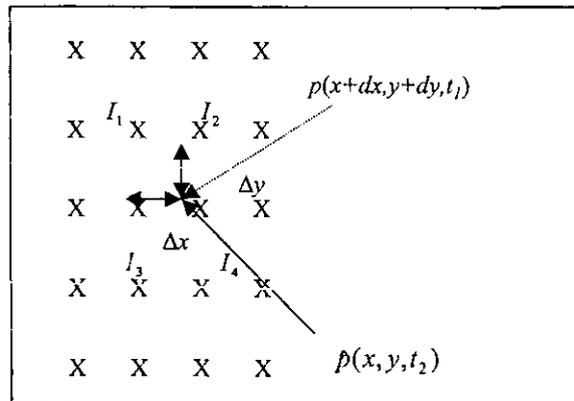


Figura A.1. Se puede observar la interpolación bilineal. El vector de desplazamiento no es un valor entero; para calcular la luminancia de un punto desplazado se utiliza la interpolación.

ANEXO B

Matrices de Rotación

Si se tiene una base $\{B\} = \{x_B, y_B, z_B\}$ y una base $\{A\} = \{x_A, y_A, z_A\}$ y están dispuestas de tal manera que sus orígenes coinciden (no necesariamente) así también uno de sus elementos coinciden (tanto en dirección como en sentido) x_B y x_A y sus otros elementos tienen disposiciones no coincidentes ya sea en dirección, sentido o ambos, es decir ver figura B.1:

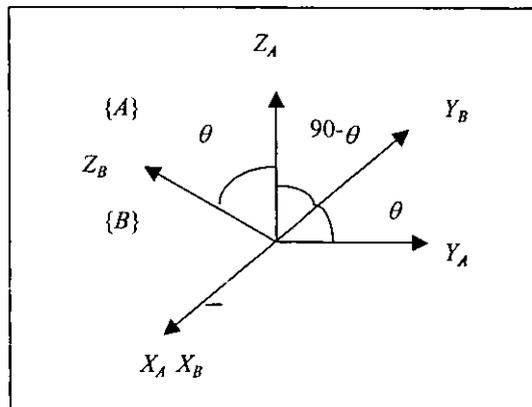


Figura B.1. Rotaciones canónicas.

De la definición de matriz de rotación, que es:

$${}^A_B R = \begin{bmatrix} (x_B, x_A) & (y_B, y_A) & (z_B, x_A) \\ (x_B, y_A) & (y_B, y_A) & (z_B, y_A) \\ (x_B, z_A) & (y_B, z_A) & (z_B, z_A) \end{bmatrix}$$

donde: (\cdot, \cdot) define el producto interno.

Así

$$(x_B, x_A) = \|x_B\| \|x_A\| \cos 0 = 1 \cdot 1 \cdot \cos 0 = 1$$

$$(x_B, y_A) = \|x_B\| \|y_A\| \cos 90^\circ = 1 \cdot 1 \cdot \cos 0 = 0$$

$$(x_B, z_A) = \|x_B\| \|z_A\| \cos 90^\circ = 1 \cdot 1 \cdot 0 = 0$$

$$(y_B, x_A) = \|y_B\| \|x_A\| \cos 90^\circ = 1 \cdot 1 \cdot 0 = 0$$

$$(y_B, y_A) = \|y_B\| \|y_A\| \cos \theta = 1 \cdot 1 \cdot \cos \theta = \cos \theta$$

$$(y_B, z_A) = \|y_B\| \|z_A\| \cos(90^\circ - \theta) = 1 \cdot 1 \cdot \cos(90^\circ - \theta) = \text{sen} \theta$$

$$(z_B, x_A) = \|z_B\| \|x_A\| \cos 90^\circ = 1 \cdot 1 \cdot 0 = 0$$

$$(z_B, y_A) = \|z_B\| \|y_A\| \cos(90^\circ + \theta) = 1 \cdot 1 \cdot \text{sen}(-\theta) = -\text{sen} \theta$$

$$(z_B, z_A) = \|z_B\| \|z_A\| \cos \theta = 1 \cdot 1 \cdot \cos \theta = \cos \theta$$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\text{sen} \theta \\ 0 & \text{sen} \theta & \cos \theta \end{bmatrix}$$

de la misma manera se pueden encontrar las matrices de rotación canónicas alrededor del eje y y z , esto es:

$$R = \begin{bmatrix} \cos \theta & 1 & \text{sen} \theta \\ 0 & 1 & 1 \\ -\text{sen} \theta & 0 & \cos \theta \end{bmatrix}$$

$$R = \begin{bmatrix} \cos\theta & -\operatorname{sen}\theta & 0 \\ \operatorname{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ANEXO C

Código de Matlab y Visual Basic

A continuación se presenta las funciones principales del código del método del gradiente ilustrado en la figura (2.4). Algunas subrutinas fueron omitidas debido a la poca relación con el algoritmo, ya que son eventos de la interface gráfica.

Matlab

```
global B;
global A;
A=imread('uno.png', 'png');
A=A';
B=imread('dos.png', 'png');
B=B';
dfd=0;
for j=154: 174,
    for i=161: 190,
        a=double(A(i,j));
        b=double(B(i,j));
        dfd=dfd+(a-b)^2;
    end
end
dfd=dfd/630;
```

```

fprintf('Inicial tx=0, ty=0, alfa=0, lambda=0, DFD=%10.3f\n',dfd);

dfd=0;
for j=154: 174,
    for i=161: 190,
        a=double(A(i,j));
        b=double(B(i+3,j-3));
        dfd=dfd+(a-b)^2;
    end
end
dfd=dfd/630;
fprintf('Compensado tx=3, ty=-3, alfa=0, lambda=0, DFD=%10.3f\n',dfd);

tx=0;
ty=0;
alfa=0;
lambda=0;

xg=174.5;
yg=163;
N=630;

for itera=1: 5,
    s1=0;
    s2=0;
    s3=0;
    s4=0;
    for j=154: 174,
        for i=161: 190,
            dx=tx+(lambda*(i-xg))-(alfa*(j-yg));
            dy=ty+(alfa*(i-xg))+(lambda*(j-yg));

            s=dif(i, j, i+dx, j+dy)/((gx(i+dx, j+dy)^2)+(gy(i+dx, j+dy)^2)+.001);
            s1=s1+(s*1*gx(i+dx, j+dy));
            s2=s2+(s*1*gy(i+dx, j+dy));
            s3x=0.01*gx(i+dx, j+dy)*(i-xg);
            s3y=0.01*gy(i+dx, j+dy)*(j-yg);
            s4x=-0.01*gx(i+dx, j+dy)*(j-yg);
            s4y=0.01*gy(i+dx, j+dy)*(i-xg);
            s3=s3+(s*(s3x+s3y));
            s4=s4+(s*(s4x+s4y));
        end
    end
    s1=s1/N;
    s2=s2/N;
    s3=s3/N;
    s4=s4/N;

```

```

tx=tx-s1;
ty=ty-s2;
alfa=alfa-s3;
lambda=lambda-s4;

dfd=0;
for j=154: 174,
    for i=161: 190,
        dx=tx+(lambda*(i-xg))-(alfa*(j-yg));
        dy=ty+(alfa*(i-xg))+(lambda*(j-yg));
        a=double(A(i,j));
        b=ip(i+dx, j+dy);
        dfd=dfd+(a-b)^2;
    end
end
dfd=dfd/630;
fprintf('i= %d, tx=%5.3f, ty=%5.3f, alfa=%5.3f, lambda=%5.3f,
DFD=%10.3f\n',itera, tx, ty, alfa, lambda, dfd);
end

```

Visual Basic

```

Private Sub mnuProcesa_Click()
mnuCopiar.Enabled = True
Screen.MousePointer = vbArrowHourglass
'Declaracion de variables
Dim xi, xf, yi, yf, j, i, itera As Integer
Dim dfd, a, b, dfdmin As Double

```

'Inicializacion de variables

```

tx = 0
ty = 0
alfa = 0
lambda = 0
itera = 1

```

'Formar la región

```

M = N * N
xi = xg - ((N - 1) / 2)
xf = xg + ((N - 1) / 2)
yi = yg - ((N - 1) / 2)

```

```
yf = yg + ((N - 1) / 2)
```

```
'Calculo el DFD natural
```

```
dfd = 0
```

```
For j = yi To yf
```

```
For i = xi To xf
```

```
  a = CDbI(FrameUno(i, j))
```

```
  b = CDbI(FrameDos(i, j))
```

```
  dfd = dfd + (a - b) ^ 2
```

```
Next i
```

```
Next j
```

```
dfd = dfd / M
```

```
MsgBox "Natural: tx=0, ty=0, alfa=0, lambda=0, DFD=" & dfd & ""
```

```
StatusBar1.Panels("ndfd").Text = "DFD:" & dfd
```

```
dfdmin = dfd
```

```
For l = -60 To 60
```

```
For k = -60 To 60
```

```
  dfd = 0
```

```
  For j = yi To yf
```

```
    If (j + 1) >= 480 Or (j + 1) <= 0 Then
```

```
      Else
```

```
        For i = xi To xf
```

```
          If (k + i) >= 640 Or (k + i) <= 0 Then
```

```
            Else
```

```
              a = CDbI(FrameUno(i, j))
```

```
              b = CDbI(FrameDos(i + k, j + 1))
```

```
              dfd = dfd + (a - b) ^ 2
```

```
            End If
```

```
          Next i
```

```
        End If
```

```
      Next j
```

```
    dfd = dfd / M
```

```
  If (dfd < dfdmin) Then
```

```
    dfdmin = dfd
```

```
    tx = k
```

```
    ty = l
```

```
  End If
```

```
Next k
```

```
Next l
```

```
MsgBox "Buscado tx=" & tx & " ,ty=" & ty & " ,alfa=" & alfa & " ,lambda=" &
lambda & " ,dfdmin=" & dfdmin & ""
```

```
'Minimizacion del DFD
```

```
Dim dx, dy, s As Double
```

```
txmin = tx
```

```
tymin = ty
```

```
alfamin = 0
```

```
lambdamin = 0
```

```
For itera = 1 To 5
```

```
uno = 0
```

```
dos = 0
```

```
tres = 0
```

```
cuatro = 0
```

```
For j = yi To yf
```

```
For i = xi To xf
```

```
dx = tx + (lambda * (i - xg)) - (alfa * (j - yg))
```

```
dy = ty + (alfa * (i - xg)) + (lambda * (j - yg))
```

```
0.01) s = dif(i, j, i + dx, j + dy) / ((gx(i + dx, j + dy) ^ 2) + (gy(i + dx, j + dy) ^ 2) +
```

```
uno = uno + (s * 1 * gx(i + dx, j + dy))
```

```
dos = dos + (s * 1 * gy(i + dx, j + dy))
```

```
terx = 0.01 * gx(i + dx, j + dy) * (i - xg)
```

```
tercery = 0.01 * gy(i + dx, j + dy) * (j - yg)
```

```
xcua = -0.01 * gx(i + dx, j + dy) * (j - xg)
```

```
ycuatro = 0.01 * gy(i + dx, j + dy) * (i - xg)
```

```
tres = tres + (s * (terx + tercery))
```

```
cuatro = cuatro + (s * (xcua + ycuatro))
```

```
Next i
```

```
Next j
```

```
uno = uno / M
```

```
dos = dos / M
```

```
tres = tres / M
```

```
cuatro = cuatro / M
```

```
tx = tx - uno
```

```
ty = ty - dos
```

```
alfa = alfa - tres
```

```
lambda = lambda - cuatro
```

```
dfd = 0
```

```
For j = yi To yf
```

```
For i = xi To xf
```

```
dx = tx + (lambda * (i - xg)) - (alfa * (j - yg))
```

```

dy = ty + (alfa * (i - xg)) + (lambda * (j - yg))
a = CDb1(FrameUno(i, j))
b = ip(i + dx, j + dy)
dfd = dfd + (a - b) ^ 2
Next i
Next j
dfd = dfd / M

' MsgBox "iteracion=" & itera & " tx=" & tx & ", ty =" & ty & ", alfa=" & alfa & ",
lambda=" & lambda & ", dfd=" & dfd & ""
StatusBar1.Panels("nitera").Text = "Iteracion:" & itera
StatusBar1.Panels("ntx").Text = "tx:" & tx
StatusBar1.Panels("nty").Text = "ty:" & ty
StatusBar1.Panels("nalfa").Text = "alfa:" & alfa
StatusBar1.Panels("nlambda").Text = "lambda:" & lambda
StatusBar1.Panels("ndfd1").Text = "DFD:" & dfd

If (dfd < dfdmin) Then
    dfdmin = dfd
    txmin = tx
    tymin = ty
    alfamin = alfa
    lambdamin = lambda
End If
' MsgBox "iteracion=" & itera & " tx=" & tx & ", ty =" & ty & ", alfa=" & alfa & ",
lambda=" & lambda & ", dfd=" & dfd & ""
Next itera
Screen.MousePointer = vbDefault

MsgBox "Minimo, tx=" & txmin & ", ty=" & tymin & ", alfa=" & alfamin & "
,lambda=" & lambdamin & ",dfd=" & dfdmin & ""
StatusBar1.Panels("nitera").Text = "Minimo"
StatusBar1.Panels("ntx").Text = "tx:" & txmin
StatusBar1.Panels("nty").Text = "ty:" & tymin
StatusBar1.Panels("nalfa").Text = "alfa:" & alfamin
StatusBar1.Panels("nlambda").Text = "lambda:" & lambdamin
StatusBar1.Panels("ndfd1").Text = "DFD:" & dfdmin

cadena = "Minimo, tx=" + Str(txmin) + ", ty=" + Str(tymin) + ", alfa=" + Str(alfamin)
+ ", lambda=" + Str(lambdamin) + ", Dfd=" + Str(dfdmin)

mov1 = xg + tx
mov2 = yg + ty
'cadena2 = "Xg+dx=" + Str(mov1) + ", Yg+dy=" + Str(mov2)

cadena2 = Str(mov1) + vbTab + Str(mov2)
mnuCopiar2.Enabled = True

```

```
Toolbar1.Buttons(8).Enabled = True
End Sub
```

```
Public Function ip(X, Y) As Double
Dim h, uno, dos, tres, cuatro As Double
Dim i, j As Integer
If (X >= 640) Or (X <= 1) Then
Else

    If (Y >= 480) Or (Y <= 1) Then
        Else

            i = Fix(X)
            j = Fix(Y)
            uno = (1 - (X - Fix(X))) * (1 - (Y - Fix(Y))) * CDbI(FrameDos(i, j))
            dos = (Y - Fix(Y)) * (1 - (X - Fix(X))) * CDbI(FrameDos(i, j + 1))
            tres = (X - Fix(X)) * (Y - Fix(Y)) * CDbI(FrameDos(i + 1, j + 1))
            cuatro = (X - Fix(X)) * (1 - (Y - Fix(Y))) * CDbI(FrameDos(i + 1, j))
            h = uno + dos + tres + cuatro
            ip = h
        End If
    End If
End Function
```

```
Public Function gx(X, Y) As Double
Dim filtro(1 To 3, 1 To 3) As Double
Dim h As Double
Dim k, j As Integer

filtro(1, 1) = -1
filtro(1, 2) = 0
filtro(1, 3) = 1
filtro(2, 1) = -1
filtro(2, 2) = 0
filtro(2, 3) = 1
filtro(3, 1) = -1
filtro(3, 2) = 0
filtro(3, 3) = 1
h = 0

For j = 1 To 3
    For k = 1 To 3

        h = h + (filtro(j, k) * ip(X - j + 1, Y - k + 1))
    ' MsgBox h
```

```

    Next k
Next j

gx = h

End Function

Public Function gy(X, Y) As Double
Dim filtro(1 To 3, 1 To 3) As Double
Dim h As Double
Dim k, j As Integer

filtro(1, 1) = -1
filtro(1, 2) = -1
filtro(1, 3) = -1
filtro(2, 1) = 0
filtro(2, 2) = 0
filtro(2, 3) = 0
filtro(3, 1) = 1
filtro(3, 2) = 1
filtro(3, 3) = 1
h = 0

For j = 1 To 3
    For k = 1 To 3

        h = h + (filtro(j, k) * ip(X - j + 1, Y - k + 1))
        MsgBox h

    Next k
Next j

gy = h

End Function

Public Function dif(x1, y1, x2, y2) As Double
Dim h As Double

h = CDBl(FrameUno(x1, y1)) - ip(x2, y2)
dif = h

End Function

Private Sub mnuDfd_Click()
Dim resultado(1 To 640, 1 To 480) As Byte

```

```
For i = 1 To 640
  For j = 1 To 480
    a = FrameUno(i, j)
    b = FrameDos(i, j)
    resta = Abs(a - b)
    resultado(i, j) = resta
  Next j
Next i
```

```
On Error GoTo HandleError
```

```
  Call fg.WriteFrame(0, 0, 0, 640, 480, resultado)
  Call fg.DrawAcquiredFrame(0, 0)
  Call fg.DestroyFrame(0)
```

```
Exit Sub
```

```
HandleError:
```

```
  MsgBox Err.Description
```

```
End Sub
```

```
Private Sub mnuDiferencia_Click()
```

```
  Dim diferencia As Double
```

```
  x1 = 10.3
```

```
  y1 = 10.3
```

```
  x2 = 15.2
```

```
  y2 = 15.2
```

```
  diferencia = dif(x1, y1, x2, y2)
```

```
  MsgBox "FrameUno( " & x1 & ", " & y1 & ")=" & FrameUno(x1, y1) & ""
```

```
  MsgBox "FrameDos( " & x2 & ", " & y2 & ")=" & FrameDos(x2, y2) & ""
```

```
  MsgBox "diferencia" & diferencia & ""
```

```
End Sub
```

BIBLIOGRAFÍA

- [1] Martínez, H. & Sotelo, F., 1999, *Metrología Dimensional utilizando visión estéreo*, Trabajo de Tesis de Licenciatura, UNAM, México.
- [2] Murat. A., 1995, *Digital video processing*, Prentice Hall, Serie Procesamiento Digital de imágenes, USA.
- [3] González, R. & Woods, R., 1992, *Digital Image Processing*, Addison-Wesley, Massachussets, USA.
- [4] Stuart, R. & Norving, P., 1998, *Inteligencia Artificial (un enfoque moderno)*, Prentice Hall, Primera Edición, México, D.F.
- [5] Jet Propulsion Laboratory, 1994, *Imaging Radar*, [en línea], Pasadena California, <<http://southport.jpl.nasa.gov/anaglyphs.html>>, [Consulta: agosto 200].
- [6] StereoGraphics Corp, 1999, *StereoGraphics Technologies Give NASA Vision on Mars*, [en línea], <<http://www.stereographics.com/Mars/nasa.htm>>, [Consulta: agosto 200].
- [7] Gould,L., *What you see in product visualization is what you get*, [en línea]: Automotive Manufacturing & Production, <<http://www.automfg.com/magazine/articles/049808>>, [Consulta: agosto 200].
- [8] Carl Zeiss, 2000, *Microcirugía tridimensional*, [en línea], <<http://www.zeiss.de/zeiss/english/home.nsf/frame>>, [Consulta: agosto 2000].

- [9] Trelease, R., 1999, *Toward Virtual Anatomy: Stereoscopic 3D Interactive Multimedia Computer Programs*, [en línea]: Department of Neurobiology, UCLA, <<http://neurobiocomp.nedsch.ucla.edu/vrx/VirtAnat1.htm>>, [Consulta: agosto 2000].
- [10] Schultz, M., 1997, *Molecules 3D*, [en línea]: Widener University, <<http://www.science.windener.edu/3dmolecules.htm>>, [Consulta: agosto 2000].
- [11] Newsletter the wave , 1998, *Deep ocean engineering*, [en línea] <<http://deepocean.com/thewave/page4.htm>>, [Consulta: agosto 2000].
- [12] The school of medicine of the University of California, [en línea], <<http://cybermed.ucsd.edu/AT/AT.html>>, [Consulta: agosto 2000]
- [13] Nicolas. N., 1992, *Hiérarchie de modeles de mouvement et méthodes d'estimation asocies. Aplication au codage de séquences d'images*, Trabajo de Tesis de Doctorado, Université de Rennes, France.
- [14] Valera, B., García, V. & Sánchez. J., 1998, *Estimación de movimiento basada en regiones*, Taller Iberoamericano de Reconocimiento de Patrones, México.
- [15] Rogers, D. & Adams, A., 1990, *Mathematical Elements for Computer Graphics*, McGraw Hill, Singapore
- [16] Cevallos, F., 2000, *Visual Basic 6 enciclopedia de Microsoft*, Alfaomega Rama, México, D.F.
- [17] Data Translation, 1999, *DT-Active Open Layer User's Manual*, Malboro, M.A.