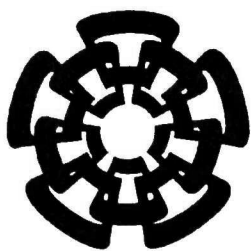




XX(132605.1)



# CINVESTAV

Centro de Investigación y de Estudios Avanzados del I.P.N.  
Unidad Guadalajara

---

## **Sistema de Realidad Aumentada**

**CINVESTAV**  
IPN  
ADQUISICIÓN  
DE LIBROS

Tesis que presenta:

**Norma Irene Serna Rodríguez**

para obtener el grado de:

**Maestro en Ciencias**

en la especialidad de:

**Ingeniería Eléctrica**

Directores de Tesis

**Dr. Raúl Ernesto González Torres**

**Dr. Luis Gerardo de la Fraga**

**CINVESTAV I.P.N.**  
**SECCION DE INFORMACION**  
**Y DOCUMENTACION**

Guadalajara, Jalisco, Noviembre de 2006.

CLASIF. TR. 165.68, 547, 2006  
ADQUISICION: SSI-445  
FECHA: 5-VII-2007  
PROCED. DON-2007  
\$ \_\_\_\_\_

I. D. 132020 - 2001

# **Sistema de Realidad Aumentada**

## **Tesis de Maestría en Ciencias Ingeniería Eléctrica**

Por:

**Norma Irene Serna Rodríguez**

Ingeniero en Sistemas Computacionales

Universidad Autónoma de Aguascalientes 1999-2004

Becario de CONACYT, expediente no. 191251

Directores de Tesis

**Dr. Raúl Ernesto González Torres**

**Dr. Luis Gerardo de la Fraga**

CINVESTAV del IPN Unidad Guadalajara, Noviembre de 2006.

# Resumen

---

Un Sistema de Realidad Aumentada (SRA) complementa el mundo real con objetos virtuales (generados por computadora) que aparecen para coexistir en el mismo espacio con el mundo real.

El propósito de esta tesis es experimentar en la creación de sistemas de realidad aumentada. Para ello se propusieron y construyeron dos sistemas de realidad aumentada: 1) un SRA en dos dimensiones en el que la idea es mover un círculo virtual sobre un plano de inclinación; el círculo virtual debe moverse de acuerdo a las leyes físicas de movimiento con fricción; y (2) un SRA en tres dimensiones que presenta el movimiento de una pelota virtual dentro de una caja real.

El sistema se construyó con una cámara de video semiprofesional conectada a una computadora vía el puerto Firewire. La aplicación fue desarrollada en C++ con las bibliotecas de Qt (para la interfaz gráfica) y Kino (para lectura de video).

El SRA consiste en reconocer dos marcas que indican el inicio y fin del plano inclinado, y para ello se aplicaron diversas técnicas de procesamiento de imagen. El SRA en tres dimensiones pertenece al área de visión por computadora en que se tiene que reconocer las esquinas de la caja y modelarla en el mundo virtual en tres dimensiones.

---

## Abstract

---

An augmented reality system (ARS) complements the real world with virtual (computer generated) objects which appear in order to coexist in a common space along with the real world.

The aim of this thesis is to experiment by constructing two augmented reality systems. For the latter, two augmented reality systems were proposed and constructed as follows: 1) A two-dimensional AR where the idea is to move a virtual circle over an inclined plane. The virtual circle has to move according to the Physics laws of movement and friction. 2) A three-dimensional AR that represents the movement of a virtual ball inside a real box.

The system was constructed by a semi-professional video camera interfaced to a computer through the Firewire port. This application was developed in C++ using the Qt (for the graphic interface) and Kino (for video driving) libraries.

The ARS recognizes two marks in order to set up the beginning and end of the inclined plane, this is done by several image processing techniques. The 3-D AR belongs to the field of computer vision and recognizes the box corners in order to translate its model to the virtual three-dimensional world.

---

# Agradecimientos

*A mis padres, Gerardo Serna Escalante e Irene Rodríguez Hernández por escucharme, apoyarme, amarme, pero sobre todo por creer en mí en todos los momentos de mi vida.*

*A mis hermanos, Gerardo, Jonathan e Ivan, por quererme, por darle vida a mi vida y hacerla totalmente fascinante.*

*A Rodrigo Jaramillo, que gracias a su comprensión, optimismo, y su muy especial forma de ver la vida han hecho que la óptica de la mía, se traduzca en el entendimiento de un mundo con mucho más color.*

*A mis asesores, el Dr. Luis Gerardo de la Fraga y el Dr. Raúl González Torres, por transmitirme los conocimientos necesarios que hicieron posible el desarrollo de este trabajo de tesis, por escucharme, entenderme y apoyarme en los momentos en que más lo necesite.*

*A mis amigos y compañeros de la Unidad Guadalajara, la Unidad Zacatenco, y a mis amigos de toda la vida, por hacer mi estancia en estos lugares más interesante, divertida e inolvidable.*

*A mi muy querida familia en "The Yellows Place", en especial a Kevin y a Drek por su guía y apoyo incondicional.*

*A CINVESTAV, a todas las personas que laboran en esta institución, en especial a las de la Unidad Guadalajara y la Unidad Zacatenco, por recibirme con la mejor sonrisa siempre.*

*A CONACyT por su interés en la actividad científica en nuestro país, ya que gracias a su apoyo hizo posible el desarrollo del presente trabajo al apoyarme con una beca y ser parcialmente apoyado por el proyecto 45306.*



*Dedicada a Dios,  
mi padre Gerardo,  
mi madre Irene,  
mis hermanitos,  
y a mi amado Kevin.*

# Índice

Portada	I
Resumen	II
Agradecimientos	IV
Dedicatoria	V
Índice de tablas	VI
Índice de figuras	IX
Índice de cuadros	X
<b>1. Introducción</b>	<b>1</b>
1.1. Definición del sistema	1
1.1.1. Sistema de realidad aumentada en dos dimensiones	2
1.1.2. Sistema de realidad aumentada en tres dimensiones	3
1.1.3. Características generales para los sistemas de realidad aumentada especificados	3
1.2. Organización de la tesis	5
1.3. Estado del arte	6
<b>2. Procesamiento de video</b>	<b>11</b>
2.1. Despliegue de video	12
2.2. Formatos de video	15
2.2.1. DV	15
2.2.2. AVI	16
2.2.3. MPEG	17
2.2.4. D-5	19
2.2.5. DVCAM	19
2.2.6. DVCPRO	19
2.2.7. Digital-S (D-9)	19
2.3. Adquisición de video	20
2.3.1. Biblioteca Kino para procesamiento de video	20
Page_Capture	22
Frame	22

---

<b>3. Procesamiento digital de marcos de video</b>	<b>27</b>
3.1. Suavizado	27
3.2. Segmentación por diferencia entre marcos	28
3.3. Segmentación por color	30
3.4. Funciones morfológicas	30
3.4.1. Erosión del marco	32
3.4.2. Dilatación del marco	32
3.5. Obtención de esquinas	32
3.5.1. Obtención de esquinas para el sistema de RA 2D	33
3.5.2. Obtención de esquinas para el sistema de RA 3D	33
<b>4. Sistema de realidad aumentada en dos dimensiones</b>	<b>37</b>
4.1. Definición del sistema	37
4.2. Creación de elementos virtuales en el marco	37
4.3. Comportamiento de un objeto en un plano inclinado con fricción	39
4.4. Resultados	41
<b>5. Sistema de realidad aumentada en tres dimensiones</b>	<b>43</b>
5.1. Definición del sistema	43
5.2. Definiciones	44
5.2.1. Modelo de la cámara	44
5.2.2. Paralelepípedos	45
5.3. Parámetros intrínsecos y extrínsecos	45
5.3.1. Matriz de proyección del paralelepípedo	46
5.3.2. Parámetros intrínsecos	50
Dualidad de parámetros intrínsecos	50
Estimación de parámetros intrínsecos de la cámara	53
5.3.3. Parámetros extrínsecos	54
Estimación de los ángulos de rotación	56
5.3.4. Coordenadas del mundo y de la imagen	57
5.4. Adición de información al marco	58
5.4.1. Objetos virtuales	58
5.5. Descripción de la interfaz	59
5.6. Movimiento físico del objeto virtual	60
5.7. Resultados	64
<b>6. Conclusiones y trabajo futuro</b>	<b>65</b>
6.1. Etapa de procesamiento de video	65
6.2. Etapa de procesamiento digital de marcos	66
6.3. Sistema de realidad aumentada en dos dimensiones	66
6.4. Sistema de realidad aumentada en tres dimensiones	66
6.5. Trabajo futuro	67

**Bibliografía**

# Índice de figuras

1.1. Sistema de realidad aumentada en dos dimensiones.	2
1.2. Sistema de realidad aumentada en tres dimensiones.	2
1.3. Modelo general del sistema de realidad aumentada.	4
2.1. Proceso de adquisición de una imagen.	12
2.2. Estructura de Formato DV.	16
3.1. Mascara de filtrado: tipo promedios.	28
3.2. Segmentación por diferencia de marcos.	30
3.3. Proceso de segmentación por color. a) marco del objeto real con líneas en color; b) marco segmentado.	31
3.4. Elemento estructural.	33
3.5. Esquinas necesarias del paralelepípedo.	34
4.1. Marco de video real.	38
4.2. Marco de video con realidad aumentada 2D.	38
4.3. a) Bloque sobre un plano inclinado. b) Diagrama de cuerpo libre del bloque.	40
4.4. Diagrama de cuerpo libre del bloque con fricción.	40
4.5. Secuencia de movimientos del objeto virtual en el sistema de RA 2D.	42
5.1. Parametrización del paralelepípedo.	45
5.2. Puntos de desvanecimiento.	47
5.3. Coordenadas paramétricas del paralelepípedo.	48
5.4. Marco de video con esquinas reconocidas, marcadas con círculos negros, y el objeto virtual dentro de la caja.	59
5.5. Interfaz del sistema.	60
5.6. Movimiento físico del objeto virtual en el sistema de realidad aumentada 3D.	62
5.7. Secuencia de movimiento del objeto virtual en el sistema de RA 3D.	63

# Índice de cuadros

2.1. Estándares de televisión a nivel mundial.	13
2.2. Formatos de video digitales y analógicos.	15

# Capítulo 1

## Introducción

---

Un sistema de Realidad Aumentada (RA) complementa el mundo real con objetos virtuales (generados por computadora), que aparecen para coexistir en el mismo espacio con el mundo real [4].

RA es una área en crecimiento dentro de la investigación en realidad virtual. El ambiente del mundo que nos rodea nos proporciona abundante información, la cual es difícil de duplicar en una computadora. Un Sistema de RA genera una vista compuesta, la cual es una combinación de escenas reales vistas por el usuario y escenas virtuales generadas por la computadora, aumentando con esto la escena con información adicional. El objetivo final es crear un sistema en el que el usuario no pueda ver la diferencia entre el mundo real y el mundo virtual aumentado de éste. Existen diversas investigaciones para crear sistemas orientados a educación, medicina, entretenimiento, entrenamiento militar, robots, telerobots, entre otros [17].

### 1.1. Definición del sistema

Se realizarán dos sistemas de realidad aumentada, uno en dos dimensiones, y otro en tres dimensiones. En ambos sistemas se tendrá un objeto real, el cual será capturado por una cámara digital de video y se mostrará en tiempo real en el monitor de la computadora, se calcularán descripciones específicas del objeto real, y de acuerdo a esta información se dibujará un objeto virtual que complementará la escena; como resultado se tendrá la visualización del objeto real y el objeto virtual interactuando en la misma escena.

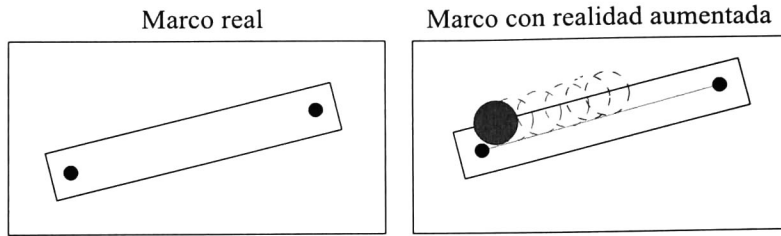


Figura 1.1: Sistema de realidad aumentada en dos dimensiones.

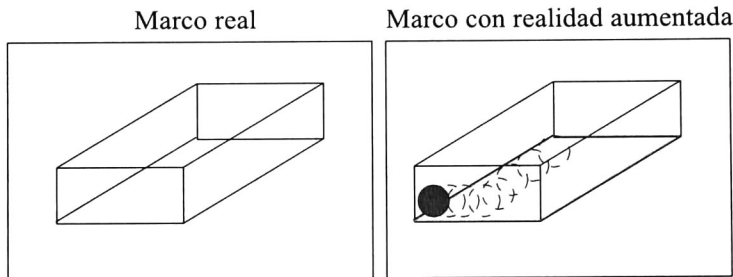


Figura 1.2: Sistema de realidad aumentada en tres dimensiones.

### 1.1.1. Sistema de realidad aumentada en dos dimensiones

El sistema de realidad aumentada en dos dimensiones consistirá en detectar dos marcas en un rectángulo (cartón rectangular), una vez detectadas estas marcas se trazará una línea entre ellas y se dibujará una pelota virtual en el centro de la línea, la pelota se debe mover como lo haría una pelota física real. Al mover el rectángulo y según su inclinación, la pelota virtual se deslizará a lo largo de la línea hasta la marca inferior, con un factor de fricción predefinido. La figura 1.1 muestra un ejemplo de una imagen (marco) y de su salida en el sistema de realidad aumentada en dos dimensiones durante un periodo de tiempo.

Un sujeto manipulará el rectángulo en el video, inclinándolo levemente hacia arriba o hacia abajo. El sistema de realidad aumentada que se realizará tomará de entrada este video, en la salida del sistema se mostrará un video en tiempo real que desplegará el video de entrada más una pelota virtual que se moverá de acuerdo a la inclinación de la línea trazada entre las marcas detectadas en el rectángulo.



### 1.1.2. Sistema de realidad aumentada en tres dimensiones

El sistema de realidad aumentada en tres dimensiones consistirá en detectar la orientación de una caja de cartón vacía, se dibujará la base de la caja con respecto a las características geométricas obtenidas, y posteriormente una pelota virtual será dibujada en el interior de la caja. Al mover la caja, y según su orientación, la pelota virtual se moverá en el interior de ella, siguiendo las leyes de movimiento con fricción de una pelota física real. La figura 1.2 muestra un ejemplo de una imagen (marco) y de su salida en el sistema de realidad aumentada en tres dimensiones durante un periodo de tiempo.

Se manipulará la caja en el video, inclinándola levemente en todos los sentidos. El sistema de realidad aumentada tomará de entrada este video, y a la salida del sistema se mostrará un video en tiempo real que desplegará el video de entrada más una pelota virtual que se moverá de acuerdo a la inclinación de la caja.

El sistema de realidad aumentada en tres dimensiones cae en el área de visión por computadora. La visión computacional es el estudio y aplicación de métodos que permiten a la computadora *entender* el contenido de las imágenes o el contenido multidimensional de datos, en donde el término entender se refiere a la extracción de información con un propósito específico. Esta área obtiene las descripciones significativas y explícitas del mundo representado por una imagen, ayuda a definir el modelo de los objetos representados en la escena, la relación entre el observador (la cámara) y la escena, así como la estructura tridimensional del espacio mostrado en ésta.

### 1.1.3. Características generales para los sistemas de realidad aumentada especificados

Para la implementación de ambos sistemas se utilizará una Cámara Canon GL2 NTSC, una Computadora Personal, y todo el software se realizará bajo la plataforma GNU/Linux, se utilizará QT y algunas bibliotecas de GTK para la implementación de la interfaz gráfica, así como especificaciones de la IEEE-1394 para la transmisión de información de video. El modelo general de este sistema se puede apreciar en la figura 1.3, cada una de las etapas que lo componen se describen más adelante.

Los sistemas de RA que se realizarán en esta tesis, procesan video mediante el uso del puerto Firewire (ieee1394) a una velocidad de 25 marcos (frames) por segundo, cada marco es extraído para ser procesado digitalmente, una vez que se obtiene el marco de video, se aplican técnicas estándar de procesamiento de imagen para poder identificar el objeto real, que en nuestro caso puede ser un par de marcas para reconocer una regla (cartón rectangular), o los bordes de la caja.

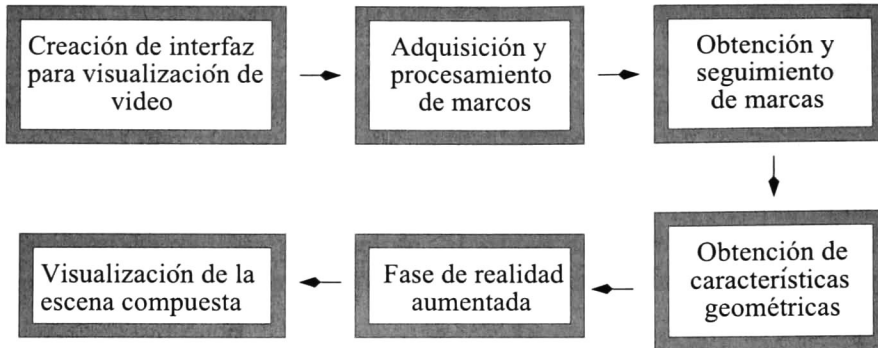


Figura 1.3: Modelo general del sistema de realidad aumentada.

Ambos sistemas de realidad aumentada tienen como función inicial detectar puntos claves, que se tomarán como referencia para entender la orientación e inclinación del objeto real en cuestión. Para el caso del rectángulo, los puntos claves son las marcas que se encuentran en los extremos del objeto. El método que se utilizará en este caso es un algoritmo de emparejamiento entre una ventana y el marco. Para el caso del paralelepípedo, dichos puntos claves representan algunas de las esquinas del objeto. El método que se utilizará es un algoritmo de segmentación por color. Posteriormente se aplicará un proceso de seguimiento de marcas (tracking).

Una vez que se ha obtenido la información necesaria que define la posición, orientación e inclinación del objeto real, entonces será posible dibujar el objeto virtual (una esfera) utilizando GTK y Qt, una vez que se ha añadido al marco el objeto virtual, se aplica una rutina que define el movimiento físico para el objeto virtual con un coeficiente de fricción, y se calcula el movimiento de éste, con base a esta información finalmente será posible conocer la posición del objeto virtual en relación al movimiento del objeto real. El diagrama a bloques que se muestra en la figura 1.3 representa las etapas básicas que deberán cubrir los sistemas de realidad aumentada especificados anteriormente, cada uno de los bloques de esta figura se explica a continuación:

- *Creación de interfaz para visualización de video.* En esta etapa se sincronizan los tiempos de despliegado de video, utilizando algunas bibliotecas de Qt, GTK y Kino [22](editor de video no lineal), así como el puerto Firewire para la transmisión de video de la cámara a la computadora. La transmisión de información se realiza a una velocidad de 25 marcos por segundo.
- *Adquisición y procesamiento de marcos.* Utilizando las bibliotecas de GTK/Kino, se extraerá directamente de memoria la información de video almacenada en un

buffer. El marco es extraído en color en formato RGB.

- *Obtención y seguimiento de marcas.* Aquí se le hará un procesamiento especial al marco, de acuerdo al sistema de realidad aumentada en cuestión será el procesamiento del marco.
- *Obtención de características geométricas.*
  - *Sistema de RA 2D.* De acuerdo a las posiciones de las marcas obtenidas, se calculará la pendiente del objeto que definirá la inclinación de éste, y en base a esto será posible definir el sistema físico que desplazará el objeto virtual hasta la marca inferior.
  - *Sistema de RA 3D.* Con base en los bordes, se debe reconocer la caja para poder dibujar el objeto virtual dentro de esta.
- *Fase de realidad aumentada.* Se construye el objeto virtual que interactuará con el objeto real, para esta tesis el objeto virtual es una esfera creada en GTK/Qt, además se establece una rutina de movimiento físico del objeto virtual.
- *Visualización de la escena compuesta.* En esta parte se despliega la imagen real con la adición del objeto virtual, y se visualiza en tiempo real la interacción entre el objeto virtual y el objeto real.

## 1.2. Organización de la tesis

Esta tesis esta organizada por 6 capítulos que describen ampliamente cada una de las fases del sistema.

Introducción. Capítulo 1. Se define el sistema que se realizará en esta tesis, un diagrama a bloques del trabajo a desarrollar, así como un resumen general de las técnicas y procesos que se emplearán en cada una de las fases del sistema; se presentan algunas definiciones sobre RA, así como la organización de la tesis en su totalidad, el estado del arte y algunos conceptos que se manejarán a lo largo de la tesis.

- *Captura de video.* Capítulo 2. Se especifican todos los pasos relacionados con el despliegue de video, lo que incluye sincronización de tiempos, paquetes y bibliotecas utilizadas, protocolos de transmisión de video y formatos de video.
- *Procesamiento digital de marcos de video.* Capítulo 3. Definición de todas las técnicas empleadas para procesar los marcos, como conversión de RGB a tonos de gris, suavizado, segmentación por diferencia de marcos, segmentación por color, rutina de detección de marcas por convolución, detección de marcas por emparejamiento, seguimiento de marcas (tracking).

- Sistema de realidad aumentada en dos dimensiones. Capítulo 4. Se explica qué información se requiere para definir la posición del objeto real y su inclinación, así como la posición inicial del objeto virtual. En base a esta información se definirá el sistema físico que establecerá el movimiento del objeto virtual. También se detalla la forma en que es creado el objeto virtual y cómo es incluido en la escena.
  
- Sistema de realidad aumentada en tres dimensiones. Capítulo 5. Detalla cómo se obtiene la aproximación de las dimensiones del paralelepípedo, mediante parámetros intrínsecos y extrínsecos de la cámara, la transformación del sistema de coordenadas del mundo al sistema de coordenadas de la imagen, la transformación de coordenadas de la imagen al sistema de coordenadas del mundo, la recuperación de las dimensiones del objeto y creación del objeto virtual, la descripción de rutina que define el movimiento físico del objeto virtual, descripción de adición del objeto virtual a la escena real, las funciones para captura de una imagen real y la captura de la imagen aumentada.

Conclusiones y trabajo futuro. Capítulo 6. Conclusiones sobre el trabajo desarrollado y propuesta de trabajo futuro.

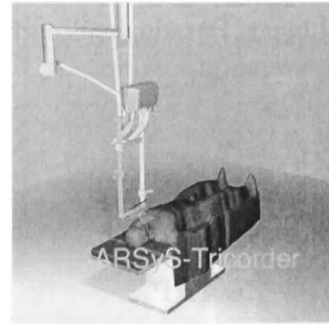
### 1.3. Estado del arte

Vivimos en un mundo saturado de información en donde es importante la forma en que apreciamos cada uno de los eventos que ocurren segundo a segundo en nuestro entorno. Las herramientas tecnológicas aunadas a todo este concepto informativo nos permiten crear múltiples visiones del mundo real, en donde además de obtener la información necesaria que satisface nuestras necesidades, podemos construir mundos alternativos interactivos que cumplan con ciertas características específicas que faciliten nuestro aprendizaje acerca del mundo desde el cual partimos originalmente. Basándonos en esta idea podemos apreciar de manera sencilla el concepto de sistemas de realidad aumentada (RA).

Algunos de los proyectos más destacados sobre RA en los últimos años son los siguientes:

**The ARSyS-Tricorder.**

AR System in Medicine (transplant design) [5].



La clínica y policlínica para cirugía maxilofacial de la Universidad de Munich esta especializada en la reconstrucción de partes de cráneo usando transplantes auto-genéticos. El objetivo del proyecto ARSyS-Tricoder es el desarrollo e integración de un sistema de realidad aumentada dentro del escenario de operación, facilitando el trabajo de reconstrucción al departamento de cirugía maxilofacial de la Universidad de Munich. Se desea proporcionar soporte durante la intervención quirúrgica mediante el sistema de realidad aumentada, el cual muestra estructuras estereoscópicas anatómicas tri-dimensionales y la ruta-quirúrgica” tridimensional dentro del escenario de operación. Además la ruta-quirúrgica no únicamente incluye la forma, volumen y posición de un tumor que será removido, sino los nervios ocultos, los vasos sanguíneos y las estructuras de tejido delgado. Este sistema ofrece la posibilidad adicional de recolectar información para un programa de operación futuro.

**ARCHEOGUIDE.**

Augmented Reality-based cultural Heritage  
On-site Guide [15].



ARCHEOGUIDE es un sistema que proveerá nuevas formas de acceso de información a sitios que son patrimonios culturales. El proyecto intenta aproximar datos culturales de una forma amigable, utilizando la integración de los avances tecnológicos en realidad aumentada, visualización-3D, computación móvil, y técnicas de interacción multimodal. El sistema tiene como objetivo inicial ser probado en la mayoría de los patrimonios culturales de Europa, dando particular énfasis a la reconstrucción virtual de ruinas.

El sistema dirigirá al usuario, ofreciéndole información de sitios culturales para visitantes, tendrá como opción la programación de sitios culturales, información sobre investigadores, y sumario de creadores. Los visitantes serán equipados con unos lentes especiales (HMD Head Mounted Display), audífonos y equipo de computación móvil. Un sistema de seguimiento determinará la ubicación del usuario con respecto

al lugar. Basado en el perfil del visitante y su posición, información visual y de audio se presentará para guiar y permitirle ganar mayor profundidad en aspectos relevantes del lugar.

### **MEDARPA.**

MEDical Augmented Reality for PATients [14].

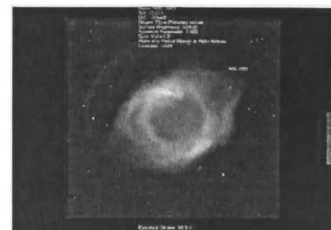


Uno de los problemas principales actualmente en las medidas de intervención quirúrgica, es la limitada visibilidad del área a tratar en el paciente. Cada área es tratada con la limitante de mínima invasión quirúrgica en cada intervención, ó no es posible visualizar el tratamiento médico; además el médico esta forzado a realizar una intervención en base a la recopilación previa de información del paciente y con la ayuda de sus experiencias en anatomía. Un problema central es la carencia de referencia espacial entre la información obtenida a priori y la posición del paciente durante la intervención.

El objetivo de MEDARPA es desarrollar un sistema de realidad aumentada-estación de trabajo de apoyo para medidas de intervención. Los componentes tecnológicos contienen nuevos procedimientos de interacción y visualización, así como mejoras de espacios de trabajo ergonómicos y el desarrollo de aparatos de despliegado de información semitransparentes. El tratamiento físico es capaz de superponerse con los datos de modalidades médicas directamente sobre el paciente, sin tener que ocupar equipo especial (HMD, cámaras) o cables restrictivos que limitan la libertad de acción.

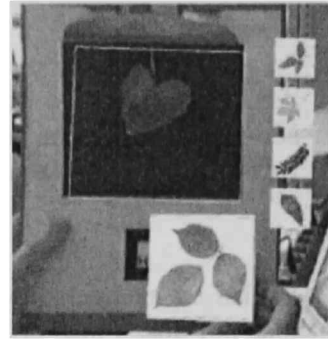
### **An Augmented Reality System for Astronomical Observations.**

MPI Informatik and TU Braunschweig, Germany. March 2006. [19]



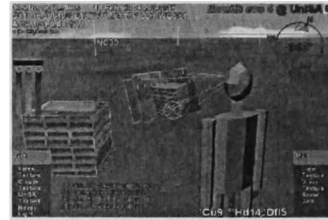
Sistema que proyecta imágenes de objetos astronómicos (enfocados en nebulosas y galaxias). Animaciones e información adicional es posible apreciarla directamente desde la vista ocular del telescopio astronómico.

**Virtual Vouchers: Prototyping a Mobile Augmented Reality User Interface for Botanical Species Identification.**  
Columbia University. March 2006. [32]



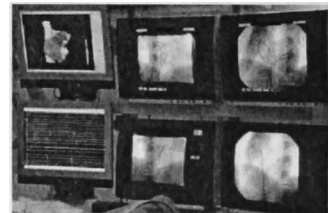
Sistema de realidad aumentada móvil que ayuda a identificar e inspeccionar especies botánicas físicas en una gran base de datos, con el objeto de determinar si se trata de una nueva especie o si ésta especie ya es conocida, al mismo tiempo que muestra características y datos de la especie encontrada en el caso de que ésta se encuentre virtualmente en la base de datos.

**3D modeling with the Tinmith mobile outdoor augmented reality system.**  
University of South Australia.  
February 2006. [23]



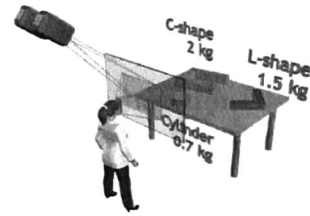
Sistema de navegación para escenarios reales que interpreta y crea objetos 3D de objetos que se encuentran actualmente en el medio ambiente, como árboles y edificios. Utiliza equipo y software especial.

**An augmented reality system for patient-specific guidance of cardiac catheter ablation procedures.**  
November 2005. [27]



Sistema que asiste en el tratamiento de arritmias cardíacas por ablación de catéter. Un modelo anatómico (3D) de un paciente dado es construido por medio de imágenes de resonancias magnéticas, el cual es mezclado con imágenes fluoroscópicas en un ambiente de RA que permite la transmisión de medidas electrocardiográficas (ECG) y actividad cardíaca en tiempo real.

**ASTOR: An auto stereoscopic optical see-through augmented reality system.**  
Royal Institute of Technology, October 2005. [21]



Utiliza un elemento holográfico transparente (HOE) para separar las vistas producidas por dos, o más proyectores digitales. No requiere que el usuario utilice lentes especiales u otro equipo; el usuario puede ver diferentes imágenes dependiendo de la posición física que tenga.

**Live three-dimensional content for augmented reality.** June 2005. [8]



Sistema de RA para superposición de contenido vivo tridimensional (3D) dentro de marcos bidimensionales en la escena. Para cada marco, la transformación Euclidiana entre marcos y cámara es estimada. La visión virtual de un modelo vivo es entonces generada y re-estructurado dentro de la escena en pasos interactivos.



# Capítulo 2

## Procesamiento de video

---

Los sistemas de procesamiento de video deben enfrentarse al desafío que supone tener que transferir, cada vez con mayor rapidez, múltiples imágenes y audio por segundo.

El video transmitido por una cámara parte del proceso de adquisición de una imagen [3] en el cual la luz reflejada por el mundo real es proyectada sobre medios ópticos. Esta imagen es enfocada hacia una superficie sensible a la luz, mientras una barra de electrones barre horizontalmente el dorso de la superficie. Esto genera una señal analógica continua. Las variaciones en la amplitud de las señales son proporcionales a la intensidad de la luz acumulada sobre la superficie sensitiva. Esta superficie es un dispositivo que transforma la energía de la luz (fotones) en electrones. Este proceso se muestra en la siguiente figura 2.1; se puede ver cómo la iluminación producida por la fuente incide sobre el objeto de la escena, en donde el lente absorbe la luz reflejada por éste convirtiéndola en señal analógica, que a su vez refleja el objeto de la escena en un plano de imagen. La señal electrónica puede ser vista como un arreglo secuencial de puntos de luz proyectados sobre una superficie sensible. El dispositivo usado para percibir la imagen es conocido como CCD (Dispositivo de cargas eléctricas Interconectadas Charge Couple Device).

### Dispositivo CCD

Un CCD es un circuito integrado que contiene un número determinado de condensadores enlazados o acoplados. Bajo el control de un circuito interno, cada condensador puede transferir su carga eléctrica a uno o a varios de los condensadores que estén a su lado en el circuito impreso. La alternativa digital a los CCD son los dispositivos CMOS (Complementary Metal Oxide Semiconductor) utilizados en algunas

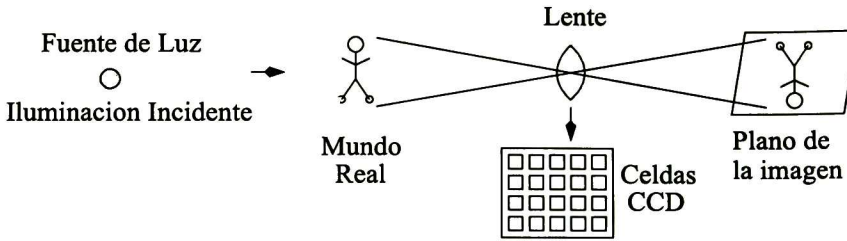


Figura 2.1: Proceso de adquisición de una imagen.

cámaras digitales y en numerosas Webcam. En la actualidad los CCD son mucho más populares en aplicaciones profesionales y en cámaras digitales.

Las funciones principales del CCD se basan en almacenamiento y exploración (scanning). Muchos dispositivos CCD contienen dos arreglos de celdas, uno para almacenar la imagen enfocada y otro que está protegido de la luz, usado para transferir la carga en el proceso de exploración. Debido a esto se dice que un CCD tiene dos modos, un modo de almacenamiento y un modo de transferencia. En un periodo activo la imagen se capta y se almacena y en un periodo de exploración se utiliza el modo de transferencia para mover la imagen a un puerto de salida.

La variación de luz obtenida hace la parte activa de la señal de video. Se agrega un pulso de sincronía al final de cada periodo activo para indicar el final de una línea barrida y el inicio de una nueva línea horizontal. Entonces, las líneas barridas o escaneadas son agrupadas para formar una imagen completa o cuadro (frame), al final del cual se introduce otra señal de sincronía para indicar el término de cada cuadro.

La tecnología de video permite manejar secuencias de imágenes que pueden ser señales de televisión o de otros dispositivos, así como de imágenes sintetizadas. Las secuencias de imágenes pueden ser capturadas y reproducidas a diferentes tasas, lo común en sistemas de cómputo son 30 cuadros por segundo. Para el procesamiento de video digital, la restricción más importante es el tiempo de captura, el cual no puede exceder  $1/30$  segundos.

## 2.1. Despliegue de video

Para la representación de colores en un dispositivo de despliegue de video (monitor, televisor, etc.) se usa un modelo que considera los tres parámetros más importantes de la luz. La luz puede ser descrita especificando su matiz de color (HUE) [9, 10],

la saturación y la luminancia. El matiz del color es su longitud de onda dominante; la luminancia está relacionada con la intensidad de la luz y la saturación es definida como el porcentaje de la intensidad de la luz que reside en el color dominante.

Para la generación de colores en video, también se usan los modelos YIQ e YUV [9, 10], los cuales son básicamente un tipo de representación alterna del modelo RGB y que comúnmente son utilizados para que la transmisión y manejo de video sea más eficiente.

Las señales de televisión y video digital son transmitidas con base en un estándar. Los estándares más comunes a nivel mundial son el NTSC (National Television System Committee), usado en América y Japón, el sistema PAL (Phase Alternation Line), principalmente usado en Europa y Asia, y el SECAM (Sequential Colour avec Memoire), usado en Francia, los países de la ex Unión Soviética y Oriente Medio. En la tabla 2.1 se muestran algunas características de estos estándares.

Cuadro 2.1: Estándares de televisión a nivel mundial.

Sistema	Cuadros/seg	Líneas por marco (Total/Activas)
Mono (s/color)	30	525/480
Color NTSC	29.97	525/480
Color PAL	25	625/580
Color SECAM	25	625/580

Las señales que genera una cámara de video son analógicas, ya que varían tanto en tiempo como en amplitud. Mientras que el video digital es un flujo discreto de números binarios que corresponden a un valor de intensidad de luz de un píxel de la imagen en un tiempo particular. Sin embargo, el video digital sólo tiene desventajas frente al video analógico, en términos del ancho de banda. Este ancho de banda se traduce en el detalle o agudeza (sharpness) de la imagen, que a su vez, en video blanco y negro se traduce en el número de líneas verticales que definen a la imagen. Para alcanzar la misma calidad de imagen, el video digital necesita mayor ancho de banda que el video analógico.

### Sistemas analógicos y digitales

El proceso de almacenamiento analógico toma la señal de video en video compuesto, extrayendo las señales de crominancia y luminancia y grabándolas en el mismo medio o sustrato; típicamente medios magnéticos u ópticos. Esto se consigue al magnetizar pequeñas partículas de metal oxidado que contienen las cintas, a través de las señales eléctricas del video. Este proceso permite el rehúso de las cintas, lo cual, aunado a lo barato de las cintas, permite el almacenamiento de video a muy bajo costo.

La desventaja que tiene es que, debido a su reuso continuo, la cinta se desgasta a consecuencia del proceso de lectura, ya que la cabeza lectora roza con la superficie de la cinta, ocasionando el degrade del material. Los problemas de estos sistemas de almacenamiento son que el video es grabado en secuencia y la recuperación de una secuencia específica es muy lenta.

Los formatos de casetes para video son muy variados. Los principales son:

- \* Cintas de una pulgada: pueden registrar 480 líneas de resolución.
- \* BetaCam, tiene cuatro pistas de audio y una de video. Actualmente existen sistemas analógicos y digitales.
- \*  $\frac{3}{4}$  U-matic: permiten grabar a una resolución de 340 líneas.
- \* S-VHS: graban a una resolución de 400 líneas.
- \* Hi8 (400 líneas).
- \* VHS: estas cintas de media pulgada pueden alcanzar a grabar con una resolución de 240 líneas.

La alternativa para el uso de las cintas son los sistemas de grabación en medios ópticos o discos láseres analógicos; los cuales graban el video en círculos concéntricos, con un solo cuadro por círculo. Este sistema permite tanto el acceso cuadro por cuadro, como efectos especiales de congelamiento de la imagen, cámara lenta, avances y retrocesos rápidos. El acceso sigue siendo en secuencias, pero la velocidad para alcanzar un cuadro específico es mucho más rápido. Este sistema permite almacenar 54000 cuadros por cada lado del disco, lo cual se traduce en 30 minutos de video en formato NTSC a 30 cuadros por segundo, por cada lado del disco.

Los dispositivos de almacenamiento digital fueron usados en primera instancia para cubrir dos requerimientos fundamentales para grabar audio y video, que son: 1) el grabado y la reproducción debe ser en tiempo real y 2) el sistema tiene que grabar o reproducir ininterrumpidamente por una hora o más. El grabado en tiempo real significa que se debe grabar a cualquier tasa que produzca la fuente de datos, lo cual no implica que se pueda grabar una señal por partes. De esta forma si una señal de video necesita una tasa de 100 Mb/s, el sistema de grabado debe grabar a esa tasa, entregando al final la grabación de la duración completa de la señal sin interrupciones. Una de las formas para que los sistemas de grabado alcanzaran esas tasas fue el recorrer la cinta de grabación a altas velocidades; la desventaja fue que sólo se lograban grabar pocos minutos. Debido a esto se propusieron técnicas que permitieran grabar a altas tasas por periodos largos.

En la actualidad hay una gran diversidad de tecnologías que ofrecen presentar video en formato digital o en formato analógico. Algunos de estos formatos se pueden apreciar en la tabla 2.2, los cuales se detallarán con más profundidad a lo largo de este capítulo.

Cuadro 2.2: Formatos de video digitales y analógicos.

Formato Digital	Formato Analógico
Digital 8	VHS
DV	SVHS
DVCAM	Betamax
DVCPRO	8mm
Betacam SP	Betacam
DigiBeta	Hi8
HDCAM	$\frac{3}{4}$ U-matic

Existe una gran variedad de algoritmos para procesar video en tiempo real, que generalmente dependen del puerto que se esté utilizando; los más comunes actualmente son: el puerto USB (Bus Serie Universal) y el puerto Firewire (estándar IEEE 1394) [2]. Este último es un puerto USB de alta velocidad por el cual se transmite audio y video en formato DV (Video Digital). Las computadoras más recientes ya traen el puerto Firewire, con lo cual se puede adquirir audio y video digital de una cámara de video con las mismas características.

En esta tesis se trabajó con el puerto Firewire y una cámara DV marca Canon GL2 [1], la cual se puede conectar directamente a una computadora a través del conector RCA (analógico) ó a través del puerto Firewire (digital) en formato DV.

## 2.2. Formatos de video

El video digital entrega mejor calidad de imágenes que el analógico; por lo tanto la reproducción de imágenes presentan una mejor definición y color. Esta es la mayor ventaja sobre el video analógico.

### 2.2.1. DV

El flujo de datos de un DV está compuesto de tres niveles [18] en una estructura jerárquica. La estructura del formato DV se muestra en la figura 2.2.

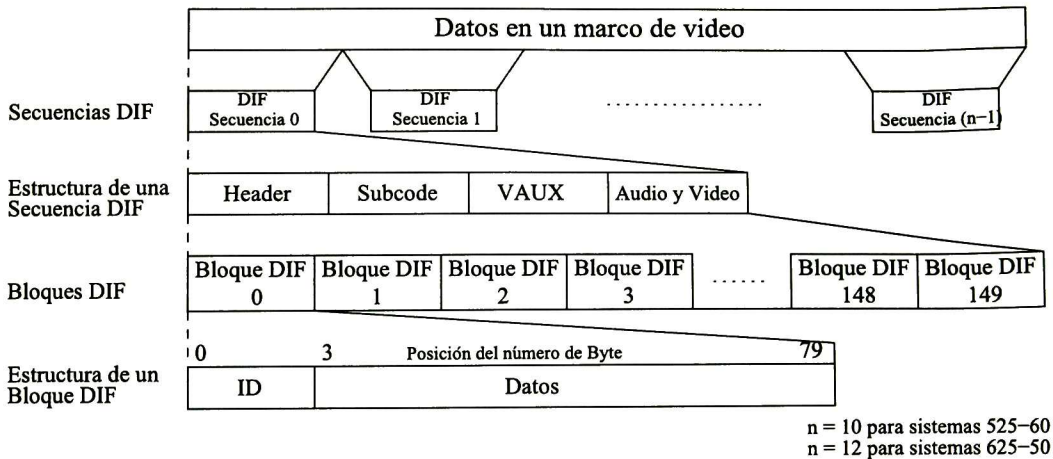


Figura 2.2: Estructura de Formato DV.

Un simple marco en el formato DV está dividido en múltiples secuencias *DIF* (*Formato de Interface Digital*). Una secuencia DIF está compuesta por 150 segmentos de 80-bytes de longitud que componen un bloque DIF. Un bloque DIF es la unidad primitiva para todo el flujo DV y es común en cualquier especificación de la familia DV. Cada bloque DIF de 80-bytes contiene una cabecera ID de 3-bytes que especifica el tipo de bloque DIF y su posición en la secuencia DIF. Existen cinco tipos de bloques DIF que están definidos en las cabeceras ID y son: Header, Subcode, Video Auxiliary Information (VAUX), Audio data y Video data. El bloque de Audio DIF además consiste de información auxiliar de audio y de datos de audio.

Este formato no presenta grado de compresión aunque es posible utilizar un códec (compresión algorítmica) para comprimir los datos si se desea. La resolución que emplea es de 720 x 576 píxeles con un nivel de color de 24 bits y un estándar de 25 marcos (frames) por segundo, aunque esto puede ser aumentado o disminuido dependiendo de la cámara de video y el programa que almacena el video.

### 2.2.2. AVI

AVI es el acrónimo de Audio Video Interleave (intercalado de audio y video). Este formato de archivo actúa como contenedor de flujo de datos de video y audio; fue desarrollado por Microsoft en 1992. Los archivos AVI son un caso especial de archivos RIFF (Resource Interchange File Format - Formato de archivos para el intercambio de recursos), un formato de propósito general para el intercambio de datos multime-

dia que fue definido por Microsoft e IBM en los años ochenta. RIFF es un clon del formato IFF inventado por Electronic Arts in 1984 para Deluxe Paint en plataforma AMIGA.

Sólo existen dos tipos generales de AVI, los basados en video para Windows y los basados en DirectShow (originalmente ActiveMovie). AVI es un formato de archivo que guarda datos codificados de diversas formas y con la ayuda de diversos CODECS (CODificadores-DECodificadoreS) que aplican diversos factores de compresión. Existe la posibilidad de almacenar los videos en un formato AVI raw (crudo o sin compresión).

Los formatos AVI basados en Video para Windows son los que actualmente forman el núcleo *dev*, los denominados AVI DV (Digital Video Video Digital) Tipo-2 mientras que los basados en DirectShow (DirectX) son los denominados AVI DV tipo-1.

Para hacer que los datos sean comprensibles para otros programas, Windows los convierte a un archivo AVI en los dos formatos comentados: Tipo-1 y Tipo-2. Microsoft ha establecido dos métodos (Tipo-1 y Tipo-2) para el almacenamiento de datos DV en archivos AVI. Aunque un AVI puede tener un número de flujos, lo más común es un flujo de vídeo (vids) y otro de audio (auds). La cabecera principal del formato AVI indica el tipo (I ó II), el número de flujos de audio y video que pueda contener el archivo, el tamaño y la resolución del video, así como el tipo de códec que utiliza para la compresión de cada flujo si es que está comprimido. El formato estándar de un AVI contempla la existencia de un flujo de video, uno o varios de audio, o ambos.

### 2.2.3. MPEG

MPEG (Moving Pictures Experts Group - Grupo de expertos de imágenes en movimiento) es parte de la Organización Internacional de Estándares (ISO), y define estándares para video digital y audio digital. La principal tarea de este grupo fue el desarrollo de un formato para reproducir video y audio en tiempo real desde un disco compacto. Todos los usos operacionales fueron cubiertos por una amplia selección de estándares, los cuales son conocidos como MPEG-1, MPEG-2, MPEG-4, MPEG-7 y MPEG-21. Cada estándar ofrece niveles y perfiles que soportan aplicaciones específicas de una forma optimizada.

Este formato ofrece tres ventajas: compatibilidad mundial, gran compresión y poca degradación en la calidad de la imagen; MPEG se compone de tres capas: audio, video y una capa a nivel de sistema, en la cual se especifica tipo de compresión, tamaño, sincronización, tiempo, calidad y razón de flujo de video.

MPEG ha establecido los siguientes formatos de compresión y normas auxiliares:

- \* MPEG-1: Establecido en 1991, es un estándar inicial de compresión de audio y video. Usado después como la norma para CD de video, incluye el popular formato de compresión de audio Capa 3 (MP3).
- \* MPEG-2: Creado en 1994, establece normas para audio y video para difusión de calidad de televisión. Utilizado para servicios de TV por satélite, señales de televisión digital por cable y (con ligeras modificaciones) para los discos de video DVD, ofrece mayor calidad con mayor ancho de banda (típicamente de 3 a 10 mega bits). En esa banda, proporciona 720x486 píxeles de resolución. Ofrece compatibilidad con MPEG-1, y tiene mayor compresión de datos. El algoritmo de compresión se basa en generar fotogramas de referencia para luego comparar los anteriores o posteriores con él y sólo las diferencias son almacenadas.
- \* MPEG-3: diseñado originalmente para HDTV (Televisión de Alta Definición), pero abandonado posteriormente en favor de MPEG-2.
- \* MPEG-4: expande MPEG-1 para soportar *objetos* audio/video, contenido 3D, codificación de baja velocidad binaria y soporte para gestión de derechos digitales (protección de copyright).
- \* MPEG-7: sistema formal para la descripción de contenido multimedia.
- \* MPEG-21: MPEG describe esta norma futura como un "marco multimedia"

El MPEG utiliza CODECS de compresión con bajas pérdidas de datos usando CODECS de transformación. En los CODECS de transformación con bajas pérdidas, las muestras tomadas de imagen y sonido son divididas en pequeños segmentos, transformadas en espacio-frecuencia y cuantificadas. Los valores cuantificados son luego codificados entrópicamente. Los sistemas de codificación de imágenes en movimiento, tal como MPEG-1, MPEG-2 y MPEG-4, añaden un paso extra, donde el contenido de imagen se predice, antes de la codificación, a partir de imágenes reconstruidas pasadas y se codifican solamente las diferencias con estas imágenes reconstruidas y algún extra necesario para llevar a cabo la predicción. MPEG solamente normaliza el formato del flujo binario y el decodificador. El codificador no está normalizado en ningún sentido, pero hay implementaciones de referencia, para los miembros, que producen flujos binarios válidos.



### 2.2.4. D-5

El D-1 fue el primer estándar digital a nivel mundial. Aún es utilizado en algunas aplicaciones especializadas de post-producción. Después siguieron D-2 y D-3, cada uno adicionando ventajas técnicas. (No existió un D-4) El D-5, uno de los más recientes, combina varias de las ventajas de formatos digitales anteriores, mientras que solventa los problemas inherentes de compresión y combinación de información de video. Los casetes D-5 graban hasta 2 horas de material en un solo videocasete. Por la posibilidad de grabar más información por unidad de tiempo, el formato D-5 puede ser adaptable a la producción DTV/HDTV. El formato D-5, junto a los Digital-S (ahora conocido como D-9) y unos cuantos formatos de alto acabado es considerado uno de los formatos digitales sin compromiso 4:2:2 (compresión mínima, luminancia y crominancia grabados separadamente y con mayor rango de muestreo digital, etc.)

### 2.2.5. DVCAM

SONY introdujo el sistema DVCAM en la convención de la NAB (Asociación Nacional de Transmisores) de 1996 como un formato digital económico. En esta línea estaba incluida una innovadora línea híbrida de edición lineal y no lineal. Las cintas DVCAM son compatibles con el formato DV

### 2.2.6. DVCPRO

El formato DVCPRO es muy popular para una gama amplia de usuarios. El editor portátil DVCPRO AJ-LT75 laptop, contiene dos unidades de reproducción, monitores a color, monitores estéreo y batería para poder operar, todo en un maletín portátil esto hizo que se convirtiera en una opción muy favorecida por reporteros. Para muchos profesionales del video, la cámara DCR-VX1000 de SONY, con su impresionante calidad digital, fue razón suficiente para cambiar sus equipos de tecnología analógica a digital. Esta cámara utiliza tres CCD en vez de uno, factor que impulsó la calidad de la primera generación del video a un rango profesional. Aunque esta calidad no puede competir con la alta calidad de los equipos 4:2:2 (de mayor precio).

### 2.2.7. Digital-S (D-9)

La calidad del formato Digital-S (D-9) excede todos los formatos analógicos e incluso muchos de los formatos digitales. Este formato ofrece cuatro canales de audio sin compresión. D-9 utiliza el diseño básico de transportación de VHS, aunque ha sido alterado (junto con la cinta y el cuerpo del casete) para ajustarse a estándares

profesionales. Aunque muchas máquinas D-9 son capaces de reproducir cintas S-VHS, el formato es muy superior al S-VHS, especialmente cuando se requieren aplicaciones multigeneracionales como la edición. Una característica importante de este formato es la pre-lectura, que permite reproducir video y audio y simultáneamente grabar nuevas señales de video y audio en su lugar. Esto significa que la señal original puede ser modificada varias veces antes de ser re-grabada. Por ejemplo, títulos y efectos especiales pueden ser adicionados mientras la cinta es reproducida.

## 2.3. Adquisición de video

### 2.3.1. Biblioteca Kino para procesamiento de video

El procesamiento de video se realizó con una cámara de video digital utilizando como salida el puerto Firewire y con formato DV. Se empleó este formato en el sistema debido a que no realiza una compresión y descompresión para obtener la información, además los flujos de audio y video se encuentran separados, facilitando la extracción de los mismos, por lo que se puede utilizar en aplicaciones de tiempo real.

Para recibir esta información de la cámara de video digital se usó la aplicación Kino. Kino es un editor de video digital con formato DV para GNU/Linux de libre distribución [22]. Sus características principales son:

- \* Integración con el IEEE-1394 (Firewire) para captura de audio y video y control VTR (Video Tape Recording).
- \* Captura de video a disco en formato DV y AVI.
- \* Control y modificación de audio y video.
- \* Control sobre grandes cantidades de video sin dividirlo.
- \* Generación de video a partir de imágenes ó de video editado.
- \* Exporta imágenes en formato ppm, jpeg, png, tiff, gif.
- \* Genera archivos de audio en formato wav, mp3 y ogg vorbis.

La aplicación Kino utiliza bibliotecas de Gnome; es posible ejecutarlo bajo diferentes escritorios, como Gnome, X Window y KDE, entre otros. La versión de kernel requerida es la 2.4.6, 2.6.12 y superiores, ya que estas versiones traen consigo el controlador del dispositivo ieeel394, necesario para la comunicación entre la cámara de video digital y la computadora.

Es importante mencionar que Kino requiere de ciertas bibliotecas para su correcto funcionamiento, las cuales se mencionan a continuación:

**gtk.** Genera y soporta interfaces gráficas.

**glib-devel.** Biblioteca de bajo nivel, contiene múltiples funciones y definiciones para desarrollar aplicaciones con GTK y GDK.

**gnome-libs.** Biblioteca para abrir, ejecutar y controlar una aplicación desarrollada con GNOME.

**x-devel.** Se utiliza para ejecutar y controlar aplicaciones con gráficos.

**imlib-devel.** Maneja diferentes formatos de imágenes.

**libxml2.** Permite manipular archivos XML.

**audiofile.** Procesa archivos de audio (aiff, aifc, wave, y NeXT/Sun).

**esound.** Demonio de sonido que permite que múltiples aplicaciones compartan una misma tarjeta de sonido.

**libraw1394.** Permite el acceso directo a la información del dispositivo que se encuentra conectado al puerto ieee1394.

**libavc1394.** Permite el control de audio y video. Esta biblioteca es utilizada como interfaz para obtener audio y video del puerto ieee1394.

**libdv.** Contiene los códecs necesarios para codificar o decodificar el formato DV.

En la realización de este trabajo de tesis, se utilizaron algunas bibliotecas de Kino para procesar el video; específicamente se empleo la forma en que Kino almacena la información que obtiene del puerto ieee1394 en memoria, para inmediatamente después desplegar dicha información a pantalla, utilizando bibliotecas de QT/GTK para generar la interfaz de despliegue de video.

Para realizar esta etapa se utilizó parte del código fuente de Kino. Las principales clases utilizadas dentro del sistema para redireccionar la información obtenida por el puerto ieee1394 mediante Kino son:

## Page\_Capture

Esta Clase contiene una función *start* que verifica el Hardware con el que cuenta nuestra computadora, comprueba niveles de seguridad y activa los procesos que darán paso a la correcta transmisión de información por el puerto Firewire. También checa preferencias de Interfaz, Canal y Buffers de captura que almacenarán en memoria la información de video, otra de las funciones incluidas en esta clase es la función *clean*, la cual destruye todos los objetos y libera los hilos inicializados en la función *start*.

## Frame

Esta clase contiene la información de audio y video por cada marco de la secuencia de video. Contiene métodos para obtener información de tamaño y contenido de los marcos. Se usó esta clase para obtener cada marco en formato RGB para posteriormente pasarlo a la etapa de procesamiento de imágenes. En esta etapa también se define la tasa de muestreo de video con la que se reproducirá. A continuación se mencionan algunas de las funciones más importantes de esta clase, así como su propósito general en el sistema.

- \* **GetFrame()**. Verifica posición del buffer de memoria y extrae una instancia de éste; dicha instancia es de tipo *Frame* con una calidad previamente especificada.
- \* **ExtractRGB(void \* rgb)**. Extrae el marco en formato RGB, cada píxel de la imagen esta basado en esta escala de colores. Este proceso lo hace verificando el tamaño de la imagen en primera instancia, después decodifica la imagen en relación a la tasa de muestreo que se especificó en la secuencia de video, y finalmente almacena en un arreglo el valor de cada píxel en sus tres componentes RGB de toda la imagen.
- \* **GetHeight()**. Obtiene la altura en píxeles del marco.
- \* **GetWidth()**. Obtiene el ancho en píxeles del marco.

Estas clases se apoyan en algunas otras bibliotecas de Kino para funcionar correctamente como son: *dv1394*, *ieee1394io* y *preferences*. Con este código es posible transmitir correctamente el video por el puerto Firewire, por lo tanto el siguiente paso es desplegar el video en sincronía con la información que va llegando al Buffer de captura. Todo este proceso se realiza con la ayuda de GTK.

El siguiente código fuente (algoritmo [1]) despliega la información recibida por el puerto Firewire a pantalla, con el uso de algunas funciones de QT y GTK. Este algoritmo se explica a continuación de acuerdo a la forma secuencial de las líneas de código:

---

**Algorithm 1** Algoritmo de Desplegado

---

```
1: gtk_init( 0,0 );
2: gdk_rgb_set_verbose( TRUE );
3: gdk_rgb_init();
4: framet = gdk_pixbuf_new( GDK_COLORSPACE_RGB, FALSE, 8, 720, 500 );
5: window = gtk_window_new( GTK_WINDOW_TOPLEVEL );
6: gtk_window_set_title( GTK_WINDOW(window), ^augmented Reality System");
7: gtk_widget_set_usize( window, Width, Height) ;
8: gtk_window_set_policy( GTK_WINDOW (window), FALSE, FALSE, FALSE );
9: gtk_signal_connect( GTK_OBJECT (window), "destroy", GTK_SIGNAL_FUNC ( destroy_cb ),
  NULL );
10: da = gtk_drawing_area_new();
11: gtk_signal_connect (GTK_OBJECT (da), "expose_event", GTK_SIGNAL_FUNC (expose_cb),
  NULL);
12: timeout_id = gtk_timeout_add (FRAME_DELAY, timeout, NULL);
13: gtk_container_add(GTK_CONTAINER (window), da);
14: gtk_widget_show (window);
15: gtk_widget_show_all (window);
16: gtk_main ();
```

---

1 - 3 : Inicialización del entorno GTK, y especificación del mapa de colores RGB a utilizar.

4 : Creación de una estructura de tipo GdkPixbuf [11], y asignación a un buffer. Dicho buffer cuenta con un sistema óptimo de salto de renglones (rowstride). En esta función se precisa el espacio de colores para la imagen, información de transparencia, número de bits por color, ancho de la imagen en píxeles y su altura.

5 : Creación de la ventana en la que se desplegará el video. Esta ventana es de tipo GtkWidget [13].

6 8 : Se establecen características de la ventana, como son: nombre de la ventana, tamaño, y definición de no redimensionabilidad.

9 : Se define el destructor de la ventana, el cual manda llamar a la función *destroy\_cb* que se encargará de terminar con todos los procesos que estén corriendo.

10 : Se crea el área de dibujo que servirá para proyectar cada marco que sea transmitido por el puerto Firewire.

11 : Esta línea es la más importante de este algoritmo, ya que es aquí donde se establece la función que estará actualizando el marco (imagen) que se visualizará en la ventana. Más adelante (en este capítulo) se presentará esta función y se explicará su desarrollo.

- 12 : Timer que define el tiempo (milisegundos) en que se actualizará el contenido de la ventana de acuerdo a la función *timeout*.
- 13 : Inclusión del área que sea ha creado para visualizar el video.
- 14 - 15 : Acción que establece mostrar la ventana y todos sus componentes.
- 16 : Función que regresa el control al código de QT.

Como ya se mencionó anteriormente uno de los pasos más importantes es la forma en que se redibuja cada marco en pantalla y el tiempo en que lo hace, esto es fundamental para poder ver un video en tiempo real. El código que se muestra en el Algoritmo 2, lee el contenido del marco y lo adecua para poder ser mostrado en nuestra ventana. La función *expose\_cb* se activa cada vez que se detecta un evento de actualización de información en el área de dibujo (*da*).

---

#### Algorithm 2 Algoritmo de Lectura de marco

---

```

1: static gint expose_cb(GtkWidget *widget, GdkEventExpose *event){
2:     gchar *pixel;
3:     int rowstride;
4:     rowstride = gdk_pixbuf_get_rowstride(frame);
5:     pixel = gdk_pixbuf_get_pixels(frame);
6:     gdk_draw_rgb_image_dither(widget->window, widget->style->black_gc, 0, 0,
7:     720, 500, GDK_RGB_DITHER_NORMAL, pixel, rowstride, 0, 0);
8:     free(pixel);
9:     return TRUE;
9: }
```

---

2 - 4 : Se definen las variables con las que se leerá y se escribirá la información.

5 : La función *gdk\_pixbuf\_get\_pixels()* hace búsquedas dentro del buffer creado previamente y definido como *frame*, en donde checa el número de bits que hay entre el inicio de un renglón y el inicio del siguiente renglón, regresando como resultado a *rowstride* la distancia que existe entre renglones. Con esta función se precisa el espacio de colores para la imagen, información de transparencia, número de bits por color, ancho de la imagen en píxeles y su altura.

6 : la función *gdk\_draw\_rgb\_image\_dither()* dibuja una imagen RGB en el área de dibujo con un ajuste para el distorsionamiento de alineación (dither alignment). Esta función emplea en primera instancia la ventana en donde se mandará la información, después el estilo del gráfico, las coordenadas (*x*, *y*) de la esquina superior izquierda del área de dibujo, las dimensiones de ésta área, el modo de modificación de disturbios en la alineación. Para este sistema se

empleó el nivel normal, la información de píxel representada por un paquete de 24-bits de datos, el *rowstride* definido anteriormente y las coordenadas offset (compensación) de distorsión de alineación.

7 8 : Se libera la memoria del puntero, y se regresa el valor de *TRUE* a la función de actualización del marco (Algoritmo [1], línea 11) informándole que el proceso se realizó con éxito.

Hasta el momento ya es posible desplegar la información de cada marco, sólo falta definir la frecuencia en que se actualizará la ventana de dibujo para tener cubierto el problema de despliegue de video. Además, tenemos que especificar qué información contendrá, es decir, si se desplegará el video con la información original, si se aplicarán métodos de Procesamiento de Imágenes ó si se añadirá información virtual (RA) [17] al marco. En el algoritmo 3 se muestra el proceso que se empleó en esta tesis.

---

### Algorithm 3 Algoritmo de Control de información del marco

---

```

1: gint timeout (gpointer data){
2: frames= getFrame();
3: if ( frames != NULL ) then
4:   gdk_threads_enter();
5:   frames→ExtractRGB( pixels );
6:   Proceamiento digital de marcos de video
7:   Obtención de características geométricas
8:   Fase de realidad aumentada
9:   gdk_pixbuf_copy_area(gdk_pixbuf_scale_simple(gdk_pixbuf_new_from_data(pixels,
   GDK_COLORSPACE_RGB, FALSE, 8, frames→GetWidth(), frames→GetHeight(),
   frames→GetWidth()*3, NULL, NULL), 720, 500, GDK_INTERP_HYPER), 0, 0, 720, 480,
   framet, 0, 0);
10:  gtk_widget_draw (da, NULL);
11:  gdk_threads_leave();
12: end if
13: return TRUE;
14: }
```

---

2 : Se extrae un frame con la función *getFrame*. Esta función forma parte de las funciones de Kino.

3 : Verifica si *frames* tiene datos, para entonces darle un seguimiento.

4 : Este macro marca el inicio de una sección critica en donde GDK y las funciones de GTK pueden ser llamadas.

5 : establece la base de colores del marco y la guarda en píxeles.

6 8 : En esta parte del algoritmo se mandarán ejecutar todas las técnicas de *Procesamiento digital de marcos de video*, algunas de ellas son: suavizado, segmentación por diferencia de marcos, segmentación por color, obtención de esquinas, morfología, etc. Cada uno de los procedimientos empleados en la mejora del marco de video se explicarán en el capítulo siguiente. En el caso de *Obtención de características geométricas* se aplicarán todos los procesos orientación e inclinación del objeto real, cálculo de los componentes de rotación y algunos otros parámetros de visión para el sistema de realidad aumentada en 3D. Finalmente en *Fase de realidad aumentada*, se aplicarán los métodos que definen el movimiento del objeto virtual, así como las funciones que dibujarán este objeto en el marco de video.

9 : la función *gdk\_pixbuf\_copy\_area()* copia una área rectangular del Pixbuff que se obtiene de la aplicación interna de la función *gdk\_pixbuf\_scale\_simple* al Pixbuff *frame* definido en el Algoritmo 1. Cuando hace completa esta tarea se activa un evento y aquí es donde se manda llamar a la función *expose\_cb*, referida en el Algoritmo 2, para mandar a dibujar a pantalla la información completa del marco y de sus procesos.

10 : En esta función se une toda la información recabada hasta el momento, la cual se desplegará en pantalla.

11 : Se finaliza la sección critica de llamada de funciones de GTK.

Este algoritmo es controlado por el timer (del que hablamos anteriormente) el cual manda llamar a la función *timeout()* cada 1/25 milisegundos, tiempo en el que se refresca el área de dibujo que presenta los marcos del video.



# Capítulo 3

## Procesamiento digital de marcos de video

El procesamiento digital de imágenes (marcos) juega un papel muy importante en la obtención de datos específicos, ya que gracias a técnicas empleadas en esta área es posible mejorar imágenes digitales para realizar interpretaciones más acertadas sobre su contenido, y en base a esto tomar decisiones de manera automática.

En este trabajo de tesis se emplearon técnicas de suavizado, segmentación por diferencia de marcos, segmentación por color, se aplicaron algunas funciones morfológicas como son: erosión y dilatación, se realizó la obtención de esquinas por mínimos cuadrados y obtención de esquinas por emparejamiento. Cada una de ellas se explica a continuación.

### 3.1. Suavizado

Existen múltiples técnicas enfocadas a mejorar la calidad de la imagen, entre ellas se encuentra la de *Filtros de Suavizado Espacial* usados para la reducción de ruido (noise) y borrosidad (blurring).

Los filtros de suavizado espacial son simplemente el promediado de los píxeles contenidos en el vecindario de la máscara de filtrado. Estos filtros también son conocidos como *filtros de promedio* que se refieren a filtros lowpass. La idea es ir reemplazando los valores de cada píxel en la imagen por el promedio de los valores de grises en el vecindario definidos por la máscara de filtrado. Este proceso resulta en una imagen con transiciones de nitidez reducidas a una imagen en niveles de gris.

Para ambos sistemas de RA se aplicó una máscara como la que se muestra en la figura 3.1, la cual consiste en una matriz de  $3 \times 3$  de unos, esta máscara es convolu-

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Figura 3.1: Mascara de filtrado: tipo promedios.

cionada sobre el marco extraído, mediante la siguiente ecuación:

$$R = \frac{1}{9} \sum_{i=1}^9 z_i \quad (3.1)$$

Al final de este proceso la imagen completa es dividida entre 9. Una máscara de  $m \times n$  tendrá una normalización constante igual a  $\frac{1}{mn}$ .

La idea es realizar un recorrido en todo el marco y evaluar si los valores encontrados en la convolución están en el rango de 0 a 255 (tonos de gris/imagen monocromática). Con esto se pretende unificar la imagen en intensidades y hacerla más uniforme, lo que ayudará eficientemente a los procesos adicionales que se le quieran hacer al marco, todo este proceso se muestra en el algoritmo 4.

---

**Algorithm 4** Algoritmo de Suavizado de marco.

---

**Require:** marco (imagen), máscara.

**Ensure:** marco suavizado.

- 1: Definición de número de píxeles de la máscara ( $pm$ ).
  - 2: **for** ( $i = 0$  to  $i \leq$  área total del marco  $i++$ ) **do**
  - 3:   Inicialización de la sumatoria entre píxeles y máscara ( $sum = 0$ ).
  - 4:   Lectura de renglón del marco ( $p_i$ ).
  - 5:   Lectura de renglón de la máscara ( $p_j$ ).
  - 6:   **for** ( $j = 0$  to  $j \leq$  columnas de la máscara  $j++$ ) **do**
  - 7:     Convolución de píxeles del marco y la máscara  $sum+ = p_i * p_j$ .
  - 8:   **end for**
  - 9:   cálculo del promedio  $val = (1/pm) * sum + 0.5$ .
  - 10:   marco suavizado =  $val$ .
  - 11: **end for**
- 

## 3.2. Segmentación por diferencia entre marcos

La segmentación es una técnica que convierte una imagen RGB ó monocromática en una imagen en blanco y negro (segmentación binaria) ó en una imagen de un sólo

color (segmentación por color). En el caso del sistema de RA 2D se decidió emplear una segmentación binaria para hacer el proceso de adquisición de información de una forma más rápida y eficiente. Para realizar este proceso se debe elegir un valor adecuado dentro de los niveles de grises (umbral), luego se comparan todos los niveles de grises, los que sean menores al umbral elegido se convertirán en negro y todos los mayores en blanco.

Por otra parte, la diferencia de marcos se basa en restar dos marcos (imágenes) extraídos en la adquisición de video, uno del otro, con la finalidad de obtener las características importantes que se encuentran en movimiento entre los dos marcos, una restricción básica es: los marcos deben de tener las mismas dimensiones (renglones y columnas).

Estos dos procesos en conjunto (segmentación y diferencia de marcos) nos proporcionarán un resultado de un marco binario con los elementos en movimientos que se presenten entre los dos marcos. El procesamiento de estas técnicas se describe en el algoritmo 5.

---

**Algorithm 5** Algoritmo de segmentación por diferencia de marcos.

---

**Require:** dos marcos.

**Ensure:** marco binario con objetos en movimiento.

```

1: Definición de umbral para segmentación. ( $um$ ).
2: for ( $i = 0$  to  $i \leq$  renglones del marco  $i++$ ) do
3:   Lectura de renglón del marco de salida ( $m_0$ ).
4:   Lectura de renglón del marco uno ( $m_1$ ).
5:   Lectura de renglón del marco dos ( $m_2$ ).
6:   for ( $j = 0$  to  $j \leq$  columnas del marco  $j++$ ) do
7:     Resta de píxeles  $val = abs(m_1 - m_2)$ 
8:     if  $val < um$  then
9:        $m_0 =$  negro (marco de salida en esa posición = negro).
10:    else
11:       $m_0 =$  blanco (marco de salida en esa posición = blanco).
12:    end if
13:  end for
14: end for

```

---

Este tipo de segmentación se aplicó al sistema de RA 2D con un umbral de 30 para realizar la segmentación. La figura 3.2 muestra un ejemplo de este proceso de forma más sencilla.

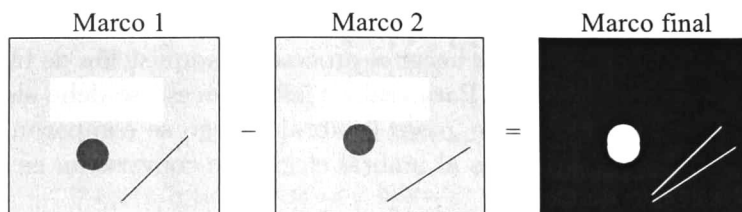


Figura 3.2: Segmentación por diferencia de marcos.

### 3.3. Segmentación por color

Para el sistema de RA 3D se utilizó este tipo de segmentación, partiendo de la obtención de un marco en el que tenemos el objeto real con marcas en color; el objetivo es segmentar correctamente estas marcas, para posteriormente ubicar las esquinas del paralelepípedo. En la figura 3.3.a se observa un ejemplo de un marco con las líneas pintadas en los extremos de la caja.

El procedimiento que se realizó para segmentar estos marcos se expresa en el algoritmo 6, el umbral que se estableció para la segmentación del sistema de RA 3D fue de 24, lo que se pretende es obtener el promedio de color en los tres componentes RGB, posteriormente se calcula una diferencia de valores entre el valor actual del componente RGB en específico y la media que se obtuvo de todo el RGB, y finalmente divide entre 3; a esto lo llamamos *varz* en el algoritmo, y este es el valor que se comparará contra el umbral para definir el color binario de la imagen resultante.

El resultado de aplicar este algoritmo a un marco se muestra en la figura 3.3 nos da como resultado.

### 3.4. Funciones morfológicas

Con la finalidad de mejorar los resultados obtenidos por la segmentación por color, se aplicó morfología matemática al marco de salida. Las funciones morfológicas que se aplicaron fueron en primer lugar erosión y posteriormente dilatación, cada una de ellas se explica a continuación.

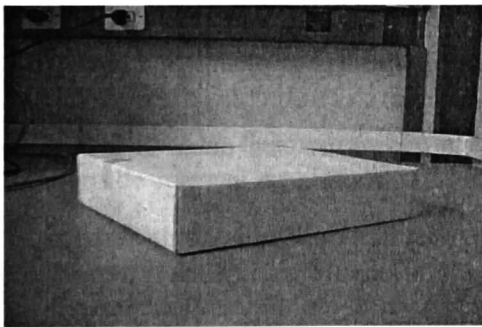
En la morfología matemática los conjuntos representan las formas de los objetos de la imagen [12]. En imágenes binarias, los conjuntos son miembros del espacio bidimensional  $Z^2$ , donde cada elemento de un conjunto es una tupla con coordenadas  $(x, y)$  de un píxel negro de una imagen.

**Algorithm 6** Algoritmo de segmentación por color.**Require:** un marco.**Ensure:** marco binario.

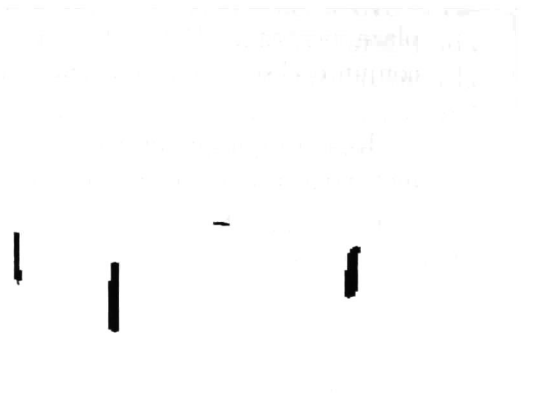
```

1: Definición de umbral para segmentación. ( $um$ ).
2: for ( $i = 0$  to  $i \leq$  renglones del marco  $i++$ ) do
3:   Lectura de renglón del marco de salida ( $m_o$ ).
4:   Lectura de renglón del marco de entrada ( $m_i$ ).
5:   for ( $j = 0$  to  $j \leq$  columnas del marco  $j++$ ) do
6:     Cálculo de la media  $media = (m_i.R + m_i.G + m_i.B)/3$ 
7:     Cálculo de la varz  $varz = abs(m_i.R - media) + abs(m_i.G - media) +$ 
       $abs(m_i.B - media)$ 
8:     if  $varz > um$  then
9:        $m_o =$  negro (marco de salida en esa posición = negro).
10:    else
11:       $m_o =$  blanco (marco de salida en esa posición = blanco).
12:    end if
13:  end for
14: end for

```



a)



b)

Figura 3.3: Proceso de segmentación por color. a) marco del objeto real con líneas en color; b) marco segmentado.

### 3.4.1. Erosión del marco

Con  $A$  y  $B$  como conjuntos de  $Z^2$ , la erosión de  $A$  por  $B$  ( $A \ominus B$ ), se define como:

$$A \ominus B = \{x | (B)_x \subseteq A\}$$

La erosión de  $A$  por  $B$  es el conjunto de todos los puntos  $x$  tales que  $B$ , trasladado por  $x$ , está contenido en  $A$ .

Una vez que hemos segmentado el marco es posible aplicarle esta operación morfológica. El efecto general de realizar erosión sobre una imagen binaria es:

1. Si todos los valores del elemento de estructura son unos, la imagen de salida tiende a ser más oscura que la imagen de entrada, y
2. se disminuye el efecto de los detalles blancos de la imagen de entrada que son menores en área que el elemento de estructura (máscara de  $3 \times 3$ ).

### 3.4.2. Dilatación del marco

Con  $A$  por  $B$  como conjuntos de  $Z^2$ , la dilatación de  $A$  por  $B$  ( $A \oplus B$ ), se define como:

$$A \oplus B = \{x | [(B^*)_x \cap A] \subseteq A\}$$

El proceso de dilatación consiste en obtener la reflexión de  $B$  sobre su origen y después cambiar esta reflexión por  $x$ . La dilatación de  $A$  por  $B$  son todos los desplazamientos  $x$  ( $B^*$ ) y  $A$  se traslapan en al menos un elemento distinto de cero. Al conjunto  $B$  se le conoce normalmente como el *elemento de estructura* de la dilatación.

Una vez que se obtiene el marco en forma binaria es posible aplicarle esta función morfológica, la cual tiene los siguientes efectos:

1. Si todos los valores del elemento de estructura son unos, la imagen de salida tiende a ser más limpia (blanca) que la de entrada, y
2. los detalles oscuros se reducen o se eliminan, dependiendo de cómo se relacionen sus valores y formas con el elemento de estructura utilizado para la dilatación, en nuestro caso se utilizó una máscara de  $3 \times 3$ .

## 3.5. Obtención de esquinas

Uno de los procesos más importantes para el correcto funcionamiento de todo el sistema es la obtención de las esquinas de nuestros objetos reales (rectángulo y paralelepípedo) que estamos utilizando.

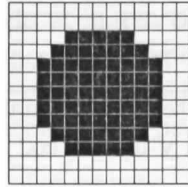


Figura 3.4: Elemento estructural.

### 3.5.1. Obtención de esquinas para el sistema de RA 2D

Para la detección de esquinas en el sistema de RA 2D se decidió pintar el rectángulo con pequeñas marcas en forma de círculos, para posteriormente identificarlas en el marco que se estuviera procesando. La obtención de esquinas (identificación de marcas) se hizo aplicando una convolución al marco en procesamiento contra un objeto estructural.

El objeto estructural es una imagen que se debe parecer a la marca que se busca, como en nuestro caso son círculos, el objeto estructural es una imagen de un círculo negro de diámetro 9 (píxeles) con borde blanco igual a 2 (píxeles) como el que se muestra en la figura [3.4].

El proceso de convolución del elemento estructural y el marco se muestran en el algoritmo 7. En este algoritmo se busca analizar pequeñas regiones del marco, y establecer una comparación entre esa región y la región del elemento estructural. Cada una de estas comparaciones tendrá un nivel de similitud, la que se calcula como la suma de los cuadrados de las diferencias entre píxeles del marco y los píxeles del elemento estructural, técnica propuesta por Smith, Sinclair, Cipolla y Wood [29].

Para encontrar las dos marcas que se necesitan en el sistema, únicamente buscamos ese número de marcas en el marco de salida y guardamos sus posiciones correspondientes.

### 3.5.2. Obtención de esquinas para el sistema de RA 3D

Para la detección de esquinas en el caso del sistema de RA 3D se analizaron algunos métodos como una diferencia de la transformada de Hough [28], obtención de aristas por SUSAN [30] y algunos otros, sin embargo los resultados que nos ofrecían estos algoritmos no cumplían con nuestras requerimientos específicos, que consisten en obtener seis de las ocho esquinas que tiene el paralelepípedo; en la figura 3.5 se muestran las esquinas que se necesitan para este sistema.

**Algorithm 7** Algoritmo de obtención de marcas.**Require:** elemento estructural y un marco.**Ensure:** posición de la marca.

```

1: for ( $i = 0$  to  $i \leq$  renglones (marco)–renglones (elemento estructural)  $i++$ ) do
2:   Lectura de renglón del marco de salida ( $m_0$ ).
3:   for ( $j = 0$  to  $j \leq$  col. del marco – col. del elemento estructural  $j++$ ) do
4:     Inicialización de la sum  $sum = 0$ .
5:     for ( $k = 0$  to  $<$  renglones del elemento estructural  $k++$ ) do
6:       Lectura de elemento estructural ( $e$ ).
7:       for ( $l = 0$  to  $<$  col. del elemento estructural  $l++$ ) do
8:         Suma del cuadrado de las diferencias entre píxeles del marco y píxeles
           del elemento estructural.
9:       end for
10:    end for
11:    Guardar valor de la diferencia de mínimos cuadrados en marco de salida.
12:  end for
13: end for
14: Encontrar mínimo en el marco de salida.
15:  $min =$  primer mínimo en el marco.
16: for ( $i = 0$  to  $i \leq$  renglones del marco  $i++$ ) do
17:   if  $min_l >$  valor actual then
18:      $min =$  valor actual.
19:     Guardar posiciones  $x, y$ .
20:   end if
21: end for

```

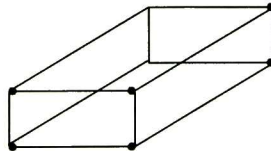


Figura 3.5: Esquinas necesarias del paralelepípedo.



Una vez que tenemos segmentado el marco como se muestra en la figura 3.3.b, y que se le han aplicado las funciones morfológicas, el siguiente paso es buscar los máximos y mínimos de cada línea, que definen las esquinas del paralelepípedo. El procedimiento que se utilizó para encontrar las posiciones de las esquinas se define en dos funciones:

*findmm*. Busca un píxel negro en el marco y manda llamar a la función *searchB()*. Esta función es llamada cada vez que se encuentre un píxel negro en el algoritmo 8.

*searchB*. Recorrer la línea definida por el píxel cuyas posiciones han sido enviadas a esta función, se avanza píxel por píxel en dirección descendente. Cada píxel que se encuentra es borrado (píxel = blanco), se guardan la posición inicial, que será la esquina superior de la línea en cuestión, y la última posición (esquina inferior) que se encontrará al terminar de recorrer la línea. Todo este proceso se puede apreciar en el algoritmo 9.

---

**Algorithm 8** Algoritmo de obtención de esquinas.

---

**Require:** marco binario.

**Ensure:** posiciones de todas las esquinas en el marco..

```

1: for ( $i = 0$  to  $i \leq$  renglones del marco  $i++$ ) do
2:   Lectura de renglón del marco de entrada ( $m_i$ ).
3:   for ( $j = 0$  to  $j \leq$  col. del marco  $j++$ ) do
4:     if  $pixm_i == negro$  then
5:       Llamar a función searchB( marco, i, j, arreglo de posiciones).
6:     end if
7:   end for
8: end for

```

---

---

**Algorithm 9** Algoritmo de recorrido.

**Require:** marco binario, coordenada del píxel encontrado  $(x, y)$ , arreglo en donde se guardarán las nuevas posiciones.

**Ensure:** posiciones de la línea.

- 1: Definir umbral de búsqueda ( $um$ ), contador = 0.
  - 2: Guardar posición inicial  $pmx_0 = x, pmy_0 = y$ .
  - 3: Borrar píxel negro inicial.
  - 4: **for** ( $j = y + 1$  to  $j \leq$  renglones del marco  $j + +$ ) **do**
  - 5:   Lectura de renglón  $y$  del marco de entrada ( $m_y$ ).
  - 6:   Buscar nuevo píxel negro en posiciones  $x - 1, x$  y  $x + 1$  del renglón leído.
  - 7:   **if**  $pix == negro$  **then**
  - 8:     Cambiar el valor del píxel a blanco, y actualizar el valor de  $y$  y  $x$ .
  - 9:   **else**
  - 10:     Incrementar contador  $contador$
  - 11:     **if**  $contador == um$  **then**
  - 12:       Salir del **for**  $break$ .
  - 13:     **end if**
  - 14:   **end if**
  - 15: **end for**
  - 16: Guardar posiciones finales  $pmx_1 = x, pmy_1 = y$ .
-

# Capítulo 4

## Sistema de realidad aumentada en dos dimensiones

### 4.1. Definición del sistema

El sistema de realidad aumentada en dos dimensiones consiste en el deslizamiento de un objeto virtual (un círculo) sobre una línea, definida en base a la detección de dos marcas. Las marcas se han obtenido de un rectángulo que es movido suavemente en tiempo real.

Los puntos importantes de este sistema son:

- \* Procesamiento de marcos de video en tiempo real.
- \* Extracción y procesamiento digital de marcos de video.
- \* Creación de los elementos virtuales en el marco de video.
- \* Comportamiento y movimiento del objeto virtual.

Los primeros dos puntos ya han sido cubiertos en los capítulos anteriores; el comportamiento del objeto virtual así como su representación se explicarán en este capítulo.

### 4.2. Creación de elementos virtuales en el marco

En este sistema se utilizan dos elementos virtuales básicos y dos opcionales; los básicos son aquellos que conforman el sistema de realidad aumentada, que para este sistema 2D son una línea y un círculo. Los dos elementos virtuales opcionales son las marcas que han sido detectadas y que definen a la línea virtual, estos elementos nos sirven únicamente para verificar la posición de la marca en la transmisión de marcos

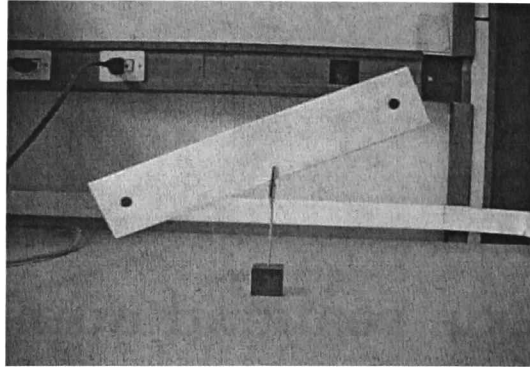


Figura 4.1: Marco de video real.

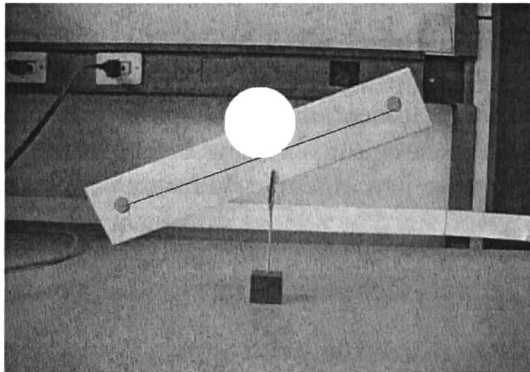


Figura 4.2: Marco de video con realidad aumentada 2D.

en tiempo real.

Una vez que se han detectado las marcas, se traza una línea recta entre ellas y se dibuja un círculo en el centro de la línea. Para definir la posición del objeto se utilizó la ecuación paramétrica de la recta:

$$l = (1 - k_x)A + k_x B \quad (4.1)$$

en donde  $k_x$  es un factor que define la posición del objeto, al inicio  $k_x = 0,5$ , para que el círculo se encuentre en el centro de la línea, después  $k_x$  será actualizado en base al comportamiento de un objeto en un plano inclinado con fricción.

Las figuras 4.1 y 4.2 muestran un marco en su extracción real y con realidad

aumentada, respectivamente. En el marco de la figura 4.2 podemos observar los elementos virtuales de los que hemos hablado en esta sección.

### 4.3. Comportamiento de un objeto en un plano inclinado con fricción

El movimiento del objeto se hizo mediante la teoría de un *bloque sobre un plano inclinado*, descrito en los libros de física Serway [25] y Resnick [26]. Este método se explica a continuación:

Un bloque de masa  $m$  es colocado sobre un plano inclinado que forma ángulo  $\theta$  con la horizontal, como se aprecia en la figura 4.3a. En la figura 4.3b se muestra el diagrama de cuerpo libre del bloque. Las fuerzas sobre él son: la normal ( $N$ ) que actúa perpendicular al plano, y el peso ( $W$ ) que actúa verticalmente hacia abajo. Por conveniencia se eligieron los ejes de coordenadas con el eje  $x$  a lo largo del plano inclinado y el eje  $y$  perpendicular a él. Al aplicar la segunda ley de Newton en la forma de componentes y observar que  $a_y = 0$ , se tiene:

$$\Sigma F_x = mg \sin \theta = ma_x \quad (4.2)$$

$$\Sigma F_y = N - mg \cos \theta = 0 \quad (4.3)$$

De las ecuaciones 4.2, 4.3 se concluye que:

- \* La aceleración a lo largo del plano inclinado está dada por la componente de peso a lo largo del plano:  $a_x = g \sin \theta$ .
- \* La componente del peso perpendicular al plano está equilibrada por la fuerza normal, tal que  $N = mg \cos \theta$ .
- \* La aceleración  $a_x$  es independiente de la masa del bloque, únicamente depende del ángulo  $\theta$  y de  $g$ .
- \* La aceleración es constante.

Para darle un enfoque más real al movimiento del objeto virtual se añadió una constante de fricción.

Al efecto producido cuando la superficie de un cuerpo se desliza sobre otro se le conoce como fuerza de fricción (los dos cuerpos ejercen una fuerza de fricción entre ellos). La fuerza de fricción de cada cuerpo es de dirección opuesta a su movimiento relativo al otro cuerpo. Las fuerzas de fricción se oponen automáticamente a este movimiento relativo y nunca contribuyen a él [26]. En esta tesis se trabajó con una

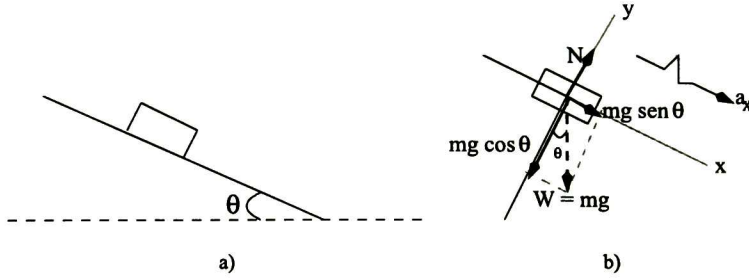


Figura 4.3: a) Bloque sobre un plano inclinado. b) Diagrama de cuerpo libre del bloque.

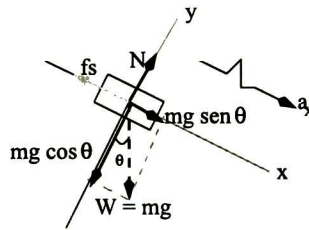


Figura 4.4: Diagrama de cuerpo libre del bloque con fricción.

fuerza de fricción estática, la cual es aquella que actúa entre superficies en reposo; una respecto a la otra.

El diagrama de cuerpo libre con una fuerza de fricción se muestra en la figura 4.4.

La razón entre la magnitud de la fuerza máxima de fricción estática y la magnitud de la fuerza normal se llama *coeficiente de fricción estática* de las superficies implicadas. Si  $f_s$  representa la magnitud de la fuerza de fricción estática, podemos escribir que:

$$f_s \leq \mu_s N \quad (4.4)$$

donde  $\mu_s$  es el coeficiente de fricción estática y  $N$  es la magnitud de la fuerza normal. El signo de igualdad se tiene sólo cuando  $f_s$  alcance su valor máximo.

La ecuación (4.2) con esta constante de fricción quedan como:

$$\Sigma F_x = mg \sin \theta - f_s = ma_x \quad (4.5)$$

De las ecuaciones [4.4, 4.5] tenemos:

$$\frac{f_s}{N} = \frac{mg \sin \theta_s}{mg \cos \theta_s} = \tan \theta_s \quad (4.6)$$

$$\mu_s = \tan \theta_s \quad (4.7)$$

El ángulo  $\theta_s$  corresponde al momento en que se inicia el desplazamiento, es el momento en donde  $f_s$  toma su valor máximo ( $f_s = \mu_s N$ ).

La aceleración del objeto ahora es: a) Cuando la pendiente es positiva [4.8]. b) Cuando la pendiente es negativa [4.9].

$$a_x = g (\mu_s \cos \theta - \sin \theta) \quad (4.8)$$

$$a_x = g (\sin \theta - \mu_s \cos \theta) \quad (4.9)$$

Como la aceleración es constante, se utilizaron las ecuaciones para velocidad y desplazamiento siguientes:

$$v = v_0 + a_x t \quad (4.10)$$

$$x = x_0 + v_0 t + \frac{1}{2} a_x t^2 \quad (4.11)$$

En cada marco se calculará el ángulo de inclinación que se tenga. En base a esto y conforme a las ecuaciones definidas anteriormente se determinará si el objeto debe moverse o no. Para el movimiento del objeto se utiliza la ecuación (4.11) incrementando o reduciendo el valor de  $k_x$ , que define la posición del objeto en  $x$  e  $y$  de la línea paramétrica.

En la figura 4.5 se muestra una secuencia de imágenes en donde se puede apreciar el movimiento del objeto virtual deslizándose en el rectángulo.

## 4.4. Resultados

El sistema de realidad aumentada en dos dimensiones nos proporcionó los siguientes resultados:

- ✱ El procesamiento de video en tiempo real es muy bueno (la transmisión de marcos se da en relación de un marco cada 30 milisegundos). Aún con los algoritmos de procesamiento de marcos y la creación de objetos virtuales, no se ve retraso en el despliegado de video.

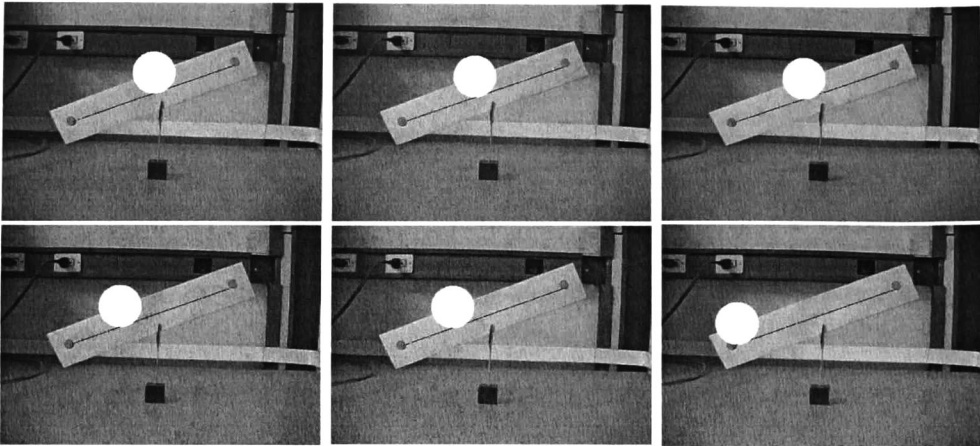


Figura 4.5: Secuencia de movimientos del objeto virtual en el sistema de RA 2D.

- \* El seguimiento de las marcas (tracking) es muy eficiente y nos permite tener una respuesta real de la posición de las marcas en todos los marcos, lo que garantiza un adecuado funcionamiento en la simulación e implementación del sistema físico que define el movimiento del objeto virtual.
- \* La creación de objetos virtuales nos permite ver claramente las posiciones de los objetos y simular eventos de una forma muy precisa y concreta. El desplazamiento del círculo que hemos empleado se desplaza perfectamente por la línea virtual, en base a todas las especificaciones que se establecieron para este sistema.



# Capítulo 5

## Sistema de realidad aumentada en tres dimensiones

### 5.1. Definición del sistema

El sistema de realidad aumentada en tres dimensiones consiste en el deslizamiento de un objeto virtual (un círculo) dentro de un paralelepípedo. Este paralelepípedo está definido en base a seis marcas (esquinas) que se han detectado del objeto real, este objeto se mueve suavemente, y empleando realidad aumentada es posible que veamos el movimiento interactivo entre el objeto virtual (círculo) y el objeto real (paralelepípedo).

Los puntos importantes de este sistema son:

- \* Procesamiento de marcos de video en tiempo real.
- \* Extracción y procesamiento digital de marcos de video.
- \* Obtención de características significativas que nos proporcionen información de la posición geométrica del objeto real.
- \* Creación de los elementos virtuales en el marco de video.
- \* Comportamiento y movimiento del objeto virtual en 3D.

En este capítulo veremos la obtención de las características geométricas importantes del objeto real, se explicará la interfaz del sistema y la creación de objetos virtuales; además se detalla como se realizó el comportamiento del objeto virtual en base a su movimiento físico con fricción, y finalmente se mencionan los resultados obtenidos.

Partiendo de imágenes que son una representación bidimensional del mundo real, se busca construir una descripción explícita de la estructura tridimensional. Dicha descripción debe incluir la identificación de objetos relevantes, así como sus características geométricas y cómo se encuentran dentro del espacio tridimensional.

Para realizar esta parte del sistema de RA 3D se investigaron varias técnicas para calibración de la cámara, entre ellas se encuentran *Uso de puntos de desvanecimiento para calibrar la cámara* de B. Caprile y V. Torre [6], *Calibración de la cámara con una sola imagen* de Zhengyou Zhang [34], *Reconstrucción de un modelo parametrizado lineal de imágenes sencillas con una cámara y distancia focal desconocida* de David Jelinek y Camillo Taylor [16], *Calibración de la cámara partiendo de puntos de desvanecimiento en imágenes de escenas arquitectónicas* de R. Cipolla, T. Drummond y D. Robertson [7] y *Calibración de la cámara y reconstrucción 3D partiendo de imágenes simples de paralelepípedos* de Wilczkowiak Marta, Boyer Edmond y Sturm Peter [20], esta última técnica se decidió aplicar a nuestro sistema de RA 3D, ya que las características se adaptan mejor a nuestro problema y los resultados son mejores respecto a las otras técnicas analizadas.

Partiendo de que se empleará la técnica de *Calibración de la cámara y reconstrucción 3D partiendo de imágenes simples de paralelepípedos*, propuesta por Wilczkowiak, Boyer y Sturm [20], empezaremos definiendo algunos conceptos importantes que se abordarán en la resolución de la obtención de las características geométricas del paralelepípedo.

## 5.2. Definiciones

### 5.2.1. Modelo de la cámara

Se asumirá que el sistema de imágenes está basado en el modelo de una cámara de agujero (que corresponde a tener una proyección en perspectiva), la proyección del espacio tridimensional al plano de la imagen bidimensional puede ser expresada por:

$$p \sim M P \quad (5.1)$$

donde  $M$  es una matriz de  $3 \times 4$ ,  $P$  es una matriz de  $4 \times n$ , ( $n$  representa el número de puntos paramétricos del paralelepípedo), y  $p$  denota las coordenadas de los puntos homogéneos en el espacio 3D y en el plano de la imagen, respectivamente. La matriz  $M$  se puede descomponer de la siguiente forma:

$$M = K [R|t] \quad (5.2)$$

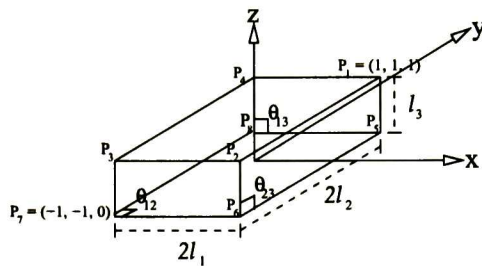


Figura 5.1: Parametrización del paralelepípedo.

$[R|t]$  es una matriz de  $3 \times 4$  que determina la orientación relativa  $R$ , y la posición  $-R^T t$  de la cámara en el espacio 3D, mientras que  $K$  es la matriz de calibración de  $3 \times 3$  que define la proyección en perspectiva de la cámara:

$$K = \begin{pmatrix} \alpha_u & s = 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (5.3)$$

donde  $\alpha_u$  y  $\alpha_v$  definen la distancia focal, expresada en dimensiones horizontal y vertical de píxeles,  $s$  es un parámetro de oblicuidad considerado igual a cero y  $u_0$ ,  $v_0$  son las coordenadas del píxel que interseca el eje óptico con el plano de la imagen.

### 5.2.2. Paralelepípedos

Un paralelepípedo está definido por doce parámetros:

- 6 parámetros extrínsecos que describen su orientación y posición en el espacio 3D (3 parámetros de posición y 3 de orientación).
- 6 parámetros intrínsecos que describen su forma Euclidiana:
  - 3 parámetros de dimensión (longitud de borde), y
  - 3 ángulos para cada uno de los ejes del paralelepípedo.

La parametrización del paralelepípedo que se empleó en este sistema de RA 3D se muestra en la figura 5.1. Los parámetros de dimensión están dados por  $l_1, l_2, l_3$ , y los ángulos que definen la unión entre cada par de ejes son:  $\theta_{12}, \theta_{13}, \theta_{23}$ .

### 5.3. Parámetros intrínsecos y extrínsecos

La obtención de los parámetros intrínsecos y extrínsecos de la cámara, proceso también conocido como *calibración de la cámara*, es una etapa muy importante ya

que nos permite construir una relación geométrica entre las coordenadas del mundo real y las coordenadas del marco.

Los algoritmos de visión por computadora para reconstruir la estructura 3D de una escena, ó calcular la posición de objetos en el espacio necesitan ecuaciones que ligen las coordenadas de los puntos en el espacio 3D con las coordenadas de sus puntos de imágenes correspondientes. Estas ecuaciones están escritas en el marco de referencia de la cámara, pero se asume frecuentemente que:

- El marco de referencia de la cámara puede estar localizado con respecto a algún otro marco de referencia conocido (el marco de referencia del mundo), y
- Las coordenadas de los puntos de la imagen en el marco de referencia de la cámara pueden obtenerse a partir de las coordenadas de los píxeles.

Esto es equivalente a suponer un conocimiento de algunas características de la cámara, conocidos en visión computacional como *parámetros intrínsecos y extrínsecos*.

Los *parámetros extrínsecos* son los parámetros que definen la localización y orientación del marco de referencia de la cámara con respecto al marco de referencia del mundo.

Los *parámetros intrínsecos* son los parámetros necesarios para ligar las coordenadas del píxel de un punto de la imagen con las coordenadas correspondientes en el marco de referencia de la cámara.

Existen muchas maneras de calibrar la cámara, entre las más conocidas está la calibración mediante puntos de desvanecimiento [6, 7, 16], cuyo objeto es expresar posteriormente la ortogonalidad del modelo. Los puntos de desvanecimiento [33] generalmente se calculan como la estimación de intersecciones de líneas (como se aprecia en la figura 5.2); sin embargo, este tipo de procedimiento genera gran inestabilidad numérica, sobre todo en casos en donde el paralelismo es muy cercano.

Para este proceso de calibración de la cámara no se calcularan explícitamente los puntos de desvanecimiento, sino que, se estimará la mejor matriz de proyección que define los puntos de la imagen correspondientes al paralelepípedo en el espacio.

### 5.3.1. Matriz de proyección del paralelepípedo

Como se mencionó en la definición de paralelepípedo, éste se compone de tres parámetros de orientación ( $R$ ), tres parámetros de posición ( $-R^T t$ ) y tres ángulos

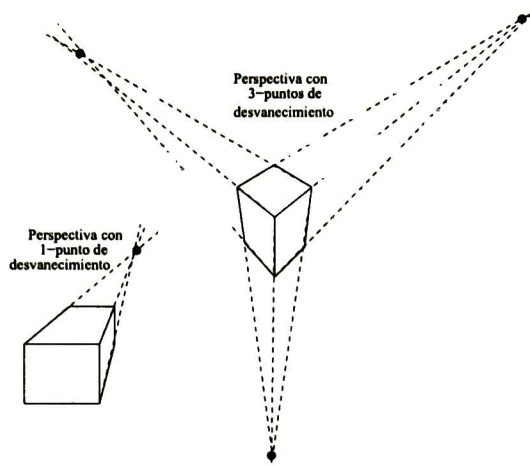


Figura 5.2: Puntos de desvanecimiento.

$(\theta_{12}, \theta_{13}, \theta_{23})$ . Las proyecciones de la imagen  $p_i = [u_i, v_i]$ , con  $i \in \{1, \dots, 8\}$ , de los vértices del paralelepípedo satisfacen la ecuación:

$$\begin{pmatrix} \alpha_1 u_1 & \alpha_2 u_2 & \cdots & \alpha_{nc} u_{cn} \\ \alpha_1 v_1 & \alpha_2 v_2 & \cdots & \alpha_{nc} v_{cn} \\ \alpha_1 & \alpha_2 & \cdots & \alpha_{cn} \end{pmatrix} = \tilde{X} \begin{pmatrix} 1 & 1 & \cdots & -1 \\ 1 & -1 & \cdots & -1 \\ 1 & 1 & \cdots & -1 \\ 1 & 1 & \cdots & 1 \end{pmatrix} \quad (5.4)$$

donde  $\tilde{X}$  es la matriz de proyección de  $3 \times 4$ , la cual está definida hasta un factor de escala por:

$$\tilde{X} \sim M \quad \Lambda \sim K [R|t] \cdot \tilde{\Lambda} \quad (5.5)$$

donde la matriz  $\tilde{\Lambda}$  depende de la parametrización que se eligió para el paralelepípedo. En este sistema se usa la siguiente:

$$\tilde{\Lambda} = \begin{pmatrix} l_1 & 0 & 0 & 0 \\ 0 & l_2 & 0 & 0 \\ 0 & 0 & l_3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.6)$$

El objetivo es obtener la matriz  $\tilde{X}$  dada por la relación que se establece en la ecuación (5.4) y la relación paramétrica de nuestro paralelepípedo, las coordenadas  $P_i$  para  $i = 1, \dots, 8$ , correspondientes a esta parametrización se pueden observar en la figura 5.3.

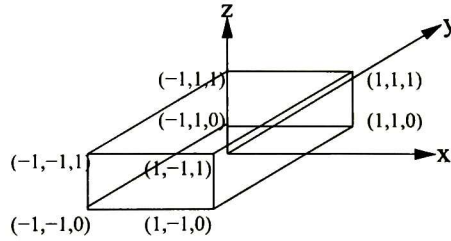


Figura 5.3: Coordenadas paramétricas del paralelepípedo.

Previamente se encontraron las esquinas de nuestro paralelepípedo como se explicó en el capítulo 3, estas coordenadas a partir de ahora las llamaremos  $[u_j, v_j]$ , en donde  $j \in 1, \dots, cn$ , que representarán los vértices  $(P_1, P_2, P_3, P_5, P_6, P_7)$  para  $cn = 6$ . Cuando  $cn = 7$  se añadirá  $P_4$ . El número de esquinas encontradas depende directamente de la vista que se tenga del paralelepípedo; su relación paramétrica se puede verificar en la figura 5.1 y su correspondiente coordenada  $(x, y, z)$  en la figura 5.3; la ecuación matricial para encontrar la matriz  $\tilde{X}$  se presentan a continuación:

$$\begin{pmatrix} \alpha_1 u_1 & \alpha_2 u_2 & \cdots & \alpha_8 u_{cn} \\ \alpha_1 v_1 & \alpha_2 v_2 & \cdots & \alpha_8 v_{cn} \\ \alpha_1 & \alpha_2 & \cdots & \alpha_{cn} \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{pmatrix} \begin{pmatrix} 1 & 1 & \cdots & -1 \\ 1 & -1 & \cdots & -1 \\ 1 & 1 & \cdots & -1 \\ 1 & 1 & \cdots & 1 \end{pmatrix}$$

Esta ecuación se reduce al sistema lineal homogéneo:

$$\begin{aligned} u_1(x_{31} + x_{32} + x_{33} + x_{34}) - x_{11} - x_{12} - x_{13} - x_{14} &= 0 \\ v_1(x_{31} + x_{32} + x_{33} + x_{34}) - x_{21} - x_{22} - x_{23} - x_{24} &= 0 \end{aligned}$$

$$\begin{aligned} u_2(x_{31} - x_{32} + x_{33} + x_{34}) - x_{11} + x_{12} - x_{13} - x_{14} &= 0 \\ v_2(x_{31} - x_{32} + x_{33} + x_{34}) - x_{21} + x_{22} - x_{23} - x_{24} &= 0 \end{aligned}$$

$$\begin{aligned} u_3(-x_{31} - x_{32} + x_{33} + x_{34}) + x_{11} + x_{12} - x_{13} - x_{14} &= 0 \\ v_3(-x_{31} - x_{32} + x_{33} + x_{34}) + x_{21} + x_{22} - x_{23} - x_{24} &= 0 \end{aligned}$$

$$\begin{aligned} u_4(-x_{31} + x_{32} + x_{33} + x_{34}) + x_{11} - x_{12} - x_{13} - x_{14} &= 0 \\ v_4(-x_{31} + x_{32} + x_{33} + x_{34}) + x_{21} - x_{22} - x_{23} - x_{24} &= 0 \end{aligned}$$

$$\begin{aligned} u_5(x_{31} + x_{32} + x_{34}) - x_{11} - x_{12} - x_{14} &= 0 \\ v_5(x_{31} + x_{32} + x_{34}) - x_{21} - x_{22} - x_{24} &= 0 \end{aligned}$$

$$\begin{aligned} u_6(x_{31} - x_{32} + x_{33} + x_{34}) - x_{11} + x_{12} - x_{14} &= 0 \\ v_6(x_{31} - x_{32} + x_{33} + x_{34}) - x_{21} + x_{22} - x_{24} &= 0 \end{aligned}$$

$$\begin{aligned} u_7(-x_{31} - x_{32} + x_{34}) + x_{11} + x_{12} - x_{14} &= 0 \\ v_7(-x_{31} - x_{32} + x_{34}) + x_{21} + x_{22} - x_{24} &= 0 \end{aligned}$$

Escrito en la forma  $Ax = 0$ , el sistema se expresa como:

$$\begin{pmatrix} -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & u_1 & u_1 & u_1 & u_1 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & v_1 & v_1 & v_1 & v_1 \\ -1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & u_2 & -u_2 & u_2 & u_2 \\ 0 & 0 & 0 & 0 & -1 & 1 & -1 & -1 & v_2 & -v_2 & v_2 & v_2 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & -u_3 & -u_3 & u_3 & u_3 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & -v_3 & -v_3 & v_3 & v_3 \\ -1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & u_5 & u_5 & 0 & u_5 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & -1 & v_5 & v_5 & 0 & v_5 \\ -1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & u_6 & -u_6 & 0 & u_6 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & -1 & v_6 & -v_6 & 0 & v_6 \\ 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & -u_7 & -u_7 & 0 & u_7 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & -v_7 & -v_7 & 0 & v_7 \\ 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & -u_4 & u_4 & u_4 & u_4 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & -1 & -v_4 & v_4 & v_4 & v_4 \end{pmatrix} \begin{pmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \\ x_{31} \\ x_{32} \\ x_{33} \\ x_{34} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (5.7)$$

La matriz de la izquierda en la ecuación (5.7) puede ser de tamaño  $(14 \times 12)$ , ó  $(12 \times 12)$ , dependiendo del número de esquinas obtenidas; los renglones que se modifican en base a esta característica son los últimos dos. Este sistema lineal homogéneo se resolvió utilizando la *descomposición del valor singular* (SVD) [24], que es una rutina para resolver problemas lineales de mínimos cuadrados basado en el teorema de álgebra lineal, el cual establece que: para cualquier matriz  $A$  de  $m \times n$ ; donde  $m \geq n$ , puede ser re-escrito como el producto de una matriz  $U$  de  $m \times n$  ortogonal por columnas, una matriz diagonal  $W$  de  $n \times n$ , con elementos cero o positivos (valores singulares), y la matriz transpuesta de una matriz ortogonal  $V$  de  $m \times n$ , como se ilustra a continuación:

$$\begin{pmatrix} A \end{pmatrix} = \begin{pmatrix} U \end{pmatrix} \begin{pmatrix} w_1 & & & \\ & \dots & & \\ & & & w_N \end{pmatrix} \begin{pmatrix} V^T \end{pmatrix}$$

Una vez que se ha calculado la matriz  $\tilde{X}$ , entonces podemos decir que tenemos toda la información geométrica dada por el paralelepípedo.

### 5.3.2. Parámetros intrínsecos

Estos pueden definirse como el conjunto de parámetros necesarios para caracterizar la óptica, la geometría y las características digitales de la cámara usada.

Para el modelo de una cámara de agujero, se necesitan tres conjuntos de parámetros intrínsecos:

- La proyección en perspectiva, para la cual sólo el largo focal  $f$  es necesario.
- La transformación entre coordenadas en el marco de la cámara y coordenadas de los píxeles.
- La distorsión geométrica introducida por la óptica.

Las coordenadas  $(x_{im}, y_{im})$  pueden pensarse como las coordenadas de un nuevo marco de referencia (coordenadas de la imagen en unidades de píxeles), algunas veces llamado *marco de referencia de la imagen*, mientras que las coordenadas  $(x_c, y_c)$ , corresponden a las coordenadas del marco de referencia de la cámara en el mismo punto.

Sin considerar las distorsiones geométricas causadas por la óptica y considerando que el ángulo CCD está compuesto por elementos fotosensibles rectangulares, se tiene:

$$\begin{aligned}x_c &= -(x_{im} - u_0) s_x \\y_c &= -(y_{im} - v_0) s_y\end{aligned}$$

con  $(u_0, v_0)$  coordenadas en píxeles del centro de la imagen (el punto principal), y  $(s_x, s_y)$  el tamaño efectivo del píxel (en milímetros) en la dirección vertical y horizontal, respectivamente. Por lo tanto, el conjunto de parámetros intrínsecos es  $f, u_0, v_0, s_x, s_y$ .

El cambio de signo se debe al hecho de que los ejes horizontal y vertical de la imagen y el marco de referencia de la cámara tienen orientaciones opuestas.

### Dualidad de parámetros intrínsecos

Consideremos la matriz  $X$  de  $3 \times 3$  definida como:

$$X \sim K \ R \cdot \Lambda \tag{5.8}$$

donde  $X$  y  $\Lambda$  son las matrices de las primeras tres líneas y columnas de  $\tilde{X}$  y  $\tilde{\Lambda}$ , respectivamente. De la ecuación (5.4) tenemos:

$$X^T \ K^{-T} \ K^{-1} \cdot X \sim \Lambda^T \ \Lambda \tag{5.9}$$



$$K^{-T} K^{-1} = w \tag{5.10}$$

$$\Lambda^T \Lambda = \mu \tag{5.11}$$

donde  $w$  es la matriz de la imagen cónica absoluta (IAC), y  $\mu$  es una entidad análoga. Ambas matrices dependen de los parámetros intrínsecos de la cámara.

La ecuación (5.9) muestra la dualidad que existe entre la cámara y un paralelepípedo; ambos tienen parámetros intrínsecos:  $K$  para la cámara y  $\Lambda$  para el paralelepípedo. Esto es importante, ya que mediante el conocimiento de alguno de los dos es posible definir restricciones que nos ayuden a encontrar la solución a los valores que no conocemos.

El siguiente paso es encontrar la matriz  $w$ , para una vez que se tengan sus valores poder conocer la matriz de calibración  $K$ . Las restricciones que consideraremos serán las siguientes:

1. Un ángulo recto  $\theta_{ij}$  nos proporciona la restricción lineal:  $X_i^T \cdot w \cdot X_j = 0$ .
2. Si se conoce la longitud de radio dada por  $r_{ij} = l_i/l_j$  entonces tendremos la restricción lineal:  $X_i^T \cdot w \cdot X_i - r_{ij}^2 X_j^T \cdot w \cdot X_j = 0$ .

Debido a que en este sistema estamos utilizando un paralelepípedo con ángulos  $\theta_{12} = \theta_{13} = \theta_{23} = 90^\circ$ , y además conocemos las dimensiones largo ( $l_2$ ), ancho ( $l_1$ ) y altura ( $l_3$ ) de la caja, podemos calcular las restricciones necesarias para obtener la matriz  $w$ .

- $\theta_{23} = 90^\circ$

$$\begin{pmatrix} x_{12} & x_{22} & x_{32} \end{pmatrix} \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix} \begin{pmatrix} x_{13} \\ x_{23} \\ x_{33} \end{pmatrix} = 0$$

$$\begin{aligned} (X_2^T \cdot w \cdot X_3) &= 0 \\ x_{13}(x_{12}w_{11} + x_{22}w_{21} + x_{32}w_{31}) + x_{23}(x_{12}w_{12} + x_{22}w_{22} + x_{32}w_{32}) + \\ & x_{33}(x_{12}w_{13} + x_{22}w_{23} + x_{32}w_{33}) = 0 \end{aligned}$$

- $\theta_{12} = 90^\circ$

$$\begin{aligned} (X_1^T \cdot w \cdot X_2) &= 0 \\ x_{12}(x_{11}w_{11} + x_{21}w_{21} + x_{31}w_{31}) + x_{22}(x_{11}w_{12} + x_{21}w_{22} + x_{31}w_{32}) + \\ & x_{32}(x_{11}w_{13} + x_{21}w_{23} + x_{31}w_{33}) = 0 \end{aligned}$$

- $\theta_{13} = 90^\circ$

$$(X_1^T \cdot w \cdot X_3) = 0$$

$$x_{13}(x_{11}w_{11} + x_{21}w_{21} + x_{31}w_{31}) + x_{23}(x_{11}w_{12} + x_{21}w_{22} + x_{31}w_{32}) +$$

$$x_{33}(x_{11}w_{13} + x_{21}w_{23} + x_{31}w_{33}) = 0$$

- $r_{12}^2 = \left(\frac{l_1}{l_2}\right)^2$

$$X_1^T \cdot w \cdot X_1 - r_{12}^2 X_2^T \cdot w \cdot X_2 = 0$$

$$(x_{11}^2 - r_{12}^2 x_{12}^2)w_{11} + (x_{11}x_{21} - x_{12}x_{22}r_{12}^2)w_{21} + (x_{31}x_{11} - x_{12}x_{32}r_{12}^2)w_{31} + (x_{21}x_{11} - x_{22}x_{12}r_{12}^2)w_{12} +$$

$$(x_{21}^2 - r_{12}^2 x_{22}^2)w_{22} + (x_{31}x_{21} - x_{22}x_{32}r_{12}^2)w_{32} + (x_{31}x_{11} - x_{32}x_{12}r_{12}^2)w_{13} + (x_{31}x_{21} - x_{22}x_{32}r_{12}^2)w_{23} +$$

$$(x_{31}^2 - r_{12}^2 x_{32}^2)w_{33} = 0$$

- $r_{13}^2 = \left(\frac{l_1}{l_3}\right)^2$

$$X_1^T \cdot w \cdot X_1 - r_{13}^2 X_3^T \cdot w \cdot X_3 = 0$$

$$(x_{11}^2 - r_{13}^2 x_{13}^2)w_{11} + (x_{11}x_{21} - x_{13}x_{23}r_{13}^2)w_{21} + (x_{31}x_{11} - x_{13}x_{33}r_{13}^2)w_{31} + (x_{21}x_{11} - x_{23}x_{13}r_{13}^2)w_{12} +$$

$$(x_{21}^2 - r_{13}^2 x_{23}^2)w_{22} + (x_{31}x_{21} - x_{23}x_{33}r_{13}^2)w_{32} + (x_{31}x_{11} - x_{33}x_{13}r_{13}^2)w_{13} + (x_{21}x_{31} - x_{33}x_{23}r_{13}^2)w_{23} +$$

$$(x_{31}^2 - r_{13}^2 x_{33}^2)w_{33} = 0$$

- $r_{23}^2 = \left(\frac{l_2}{l_3}\right)^2$

$$X_2^T \cdot w \cdot X_2 - r_{23}^2 X_3^T \cdot w \cdot X_3 = 0$$

$$(x_{12}^2 - r_{23}^2 x_{13}^2)w_{11} + (x_{12}x_{22} - x_{13}x_{23}r_{23}^2)w_{21} + (x_{12}x_{32} - x_{13}x_{33}r_{23}^2)w_{31} + (x_{22}x_{12} - x_{23}x_{13}r_{23}^2)w_{12} +$$

$$(x_{22}^2 - r_{23}^2 x_{23}^2)w_{22} + (x_{22}x_{32} - x_{23}x_{33}r_{23}^2)w_{32} + (x_{32}x_{12} - x_{33}x_{13}r_{23}^2)w_{13} + (x_{32}x_{22} - x_{33}x_{23}r_{23}^2)w_{23} +$$

$$(x_{32}^2 - r_{23}^2 x_{33}^2)w_{33} = 0$$

De acuerdo a la ecuación (5.10) tenemos lo siguiente:

$$w = K^{-T} K^{-1} = \begin{pmatrix} \frac{1}{f} & 0 & 0 \\ 0 & \frac{1}{f} & 0 \\ \frac{-u_0}{f} & \frac{-v_0}{f} & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{f} & 0 & \frac{-u_0}{f} \\ 0 & \frac{1}{f} & \frac{-v_0}{f} \\ 0 & 0 & 1 \end{pmatrix}$$

por lo que  $w$  resulta ser:

$$w = \begin{pmatrix} \frac{1}{f^2} & 0 & \frac{-u_0}{f^2} \\ 0 & \frac{1}{f^2} & \frac{-v_0}{f^2} \\ \frac{-u_0}{f^2} & \frac{-v_0}{f^2} & \frac{u_0^2 + v_0^2 + f^2}{f^2} \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix} \quad (5.12)$$

De la ecuación (5.12) podemos observar que únicamente tenemos cuatro incógnitas  $\frac{1}{f^2}$ ,  $\frac{-u_0}{f^2}$ ,  $\frac{-v_0}{f^2}$  y  $\frac{u_0^2 + v_0^2 + f^2}{f^2}$ , correspondientes a  $w_{11}$ ,  $w_{13}$ ,  $w_{23}$ ,  $w_{33}$ , respectivamente, por lo que se tienen las restricciones siguientes:

1.  $w_{11} = w_{22}$
2.  $w_{12} = w_{21} = 0$
3.  $w_{13} = w_{31}$
4.  $w_{23} = w_{32}$

En base a estas restricciones y a las que calculamos con la definición de ángulos, así como las tres ecuaciones que involucran la longitud de radio, se construye el sistema lineal homogéneo que nos permitirá encontrar los valores de la matriz  $w$ . El sistema lineal homogéneo resultante es el siguiente:

$$\begin{pmatrix} x_{13}x_{12} + x_{23}x_{22} & x_{13}x_{32} + x_{33}x_{12} & x_{23}x_{32} + x_{33}x_{22} & x_{33}x_{32} \\ x_{12}x_{11} + x_{22}x_{21} & x_{12}x_{31} + x_{32}x_{11} & x_{22}x_{31} + x_{32}x_{21} & x_{32}x_{31} \\ x_{13}x_{11} + x_{23}x_{21} & x_{13}x_{31} + x_{33}x_{11} & x_{23}x_{31} + x_{33}x_{21} & x_{33}x_{31} \\ x_{11}^2 + x_{21}^2 - (x_{12}^2 + x_{22}^2)r_{12}^2 & 2(x_{31}x_{11} - x_{32}x_{12}r_{12}^2) & 2(x_{31}x_{21} - x_{32}x_{22}r_{12}^2) & x_{31}^2 - x_{32}^2r_{12}^2 \end{pmatrix} \begin{pmatrix} w_{11} \\ w_{13} \\ w_{23} \\ w_{33} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (5.13)$$

Este sistema lineal homogéneo se resolvió utilizando la rutina del SVD [24] del libro Numerical Recipese in C++.

### Estimación de parámetros intrínsecos de la cámara

Partiendo de la ecuación (5.12), hacemos las igualaciones correspondientes y obtenemos los parámetros de la matriz de calibración  $k$ .

Longitud focal.

$$\frac{1}{f^2} = w_{11}$$

Despejando  $f$  queda como:

$$f = \sqrt{\frac{1}{w_{11}}} \quad (5.14)$$

Punto de proyección central para el eje x:

$$\frac{-u_0}{f^2} = w_{13}$$

Despejando  $u_0$  y sustituyendo la ecuación (5.14) al cuadrado, resulta:

$$u_0 = \frac{-w_{13}}{w_{11}} \quad (5.15)$$

Punto de proyección central para el eje y:

$$\frac{-v_0}{f^2} = w_{23}$$

Despejando  $v_0$  y sustituyendo la ecuación (5.14) al cuadrado, resulta:

$$v_0 = \frac{-w_{23}}{w_{11}} \quad (5.16)$$

Y con esto hemos obtenido los parámetros intrínsecos de la cámara.

### 5.3.3. Parámetros extrínsecos

El marco de referencia de la cámara se ha introducido con el propósito de escribir las ecuaciones fundamentales de la proyección en perspectiva en una forma simple. Sin embargo, el marco de referencia de la cámara es frecuentemente desconocido, y un problema común es determinar la localización y orientación del marco de la cámara con respecto a algún marco de referencia conocido, usando sólo información de la imagen.

Típicamente se escoge lo siguiente para describir la transformación entre los marcos de la cámara y el mundo:

Un vector de translación 3D,  $t$ , que describe la posición relativa de los orígenes de los dos marcos de referencia.

- Una matriz de rotación  $R$  de  $3 \times 3$ , que es una matriz ortogonal que lleva los ejes correspondientes de los dos marcos, uno sobre el otro.

Partiendo de la ecuación (5.8)  $R$  puede ser calculada así:

$$\begin{aligned} X &\sim K \cdot R \cdot \Lambda \\ X &= \lambda \cdot K \cdot R \cdot \Lambda \\ K^{-1}X &= \lambda \cdot K^{-1}K \cdot R \cdot \Lambda \\ K^{-1}X &= \lambda \cdot R \cdot \Lambda \\ K^{-1}X\Lambda^{-1} &= \lambda \cdot R \cdot \Lambda\Lambda^{-1} \\ K^{-1}X\Lambda^{-1} &= \lambda \cdot R \\ R &= \frac{1}{\lambda}K^{-1}X\Lambda^{-1} \end{aligned}$$

La matriz de rotación es entonces:

$$\begin{aligned} R &= \frac{1}{\lambda}K^{-1}X\Lambda^{-1} \quad (5.17) \\ &= \frac{1}{\lambda} \begin{pmatrix} \frac{1}{f} & 0 & \frac{-u_0}{f} \\ 0 & \frac{1}{f} & \frac{-v_0}{f} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} \begin{pmatrix} \frac{1}{l_1} & 0 & 0 \\ 0 & \frac{1}{l_2} & 0 \\ 0 & 0 & \frac{1}{l_3} \end{pmatrix} \end{aligned}$$

$$= \frac{1}{\lambda} \begin{pmatrix} \frac{x_{11}-u_0x_{31}}{l_1f} & \frac{x_{12}-u_0x_{32}}{l_2f} & \frac{x_{13}-u_0x_{33}}{l_3f} \\ \frac{x_{21}-v_0x_{31}}{l_1f} & \frac{x_{22}-v_0x_{32}}{l_2f} & \frac{x_{23}-v_0x_{33}}{l_3f} \\ \frac{x_{31}}{l_1} & \frac{x_{32}}{l_2} & \frac{x_{33}}{l_3} \end{pmatrix}$$

Suponiendo  $\lambda = 1$  la matriz de rotación queda como:

$$R = \begin{pmatrix} \frac{x_{11}-u_0x_{31}}{l_1f} & \frac{x_{12}-u_0x_{32}}{l_2f} & \frac{x_{13}-u_0x_{33}}{l_3f} \\ \frac{x_{21}-v_0x_{31}}{l_1f} & \frac{x_{22}-v_0x_{32}}{l_2f} & \frac{x_{23}-v_0x_{33}}{l_3f} \\ \frac{x_{31}}{l_1} & \frac{x_{32}}{l_2} & \frac{x_{33}}{l_3} \end{pmatrix} \quad (5.18)$$

Como ya tenemos todos los parámetros que aparecen en la ecuación (5.18), entonces ya podemos conocer explícitamente los valores de la matriz de rotación. Con la finalidad de mejorar los resultados de esta matriz, normalizamos sus valores y forzamos las restricciones de ortogonalidad con la ayuda de la rutina SVD [31].

La matriz  $R$  se puede descomponer como:  $R_{3 \times 3} = U_{3 \times 3} D_{3 \times 3} V_{3 \times 3}^T$ , donde las columnas de las matrices  $U$  y  $V$  son vectores unitarios mutuamente ortogonales, y la matriz  $D$  es una matriz diagonal, cuyos elementos llamados  $d_{ii}$  son *valores singulares*, tales que  $d_{ii} \geq 0$ . Como los valores de  $U$  y  $V$  no son únicos, los valores  $d_{ii}$  son totalmente determinados por  $R$ . Para asegurar la condición de ortogonalidad debemos forzar a que  $d_{ii} = 1$ . Para realizar este procedimiento utilizamos la rutina de SVD, descomponiendo la matriz  $R$  mediante la función `svdcmp( R, columnas, renglones, D, V)`. Esta función descompondrá la matriz en las matrices  $U, V$  y  $D$ , mencionadas anteriormente. Una vez que tengamos esta descomposición, utilizaremos la transpuesta matriz resultante  $V$  y definiremos  $D$  en base a  $d_{ii} = 1$ , finalmente obtendremos la nueva matriz  $R$  de la siguiente forma:

$$\tilde{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \\ v_{13} & v_{23} & v_{33} \end{pmatrix} \quad (5.19)$$

Después solo actualizaremos la matriz de rotación reemplazando  $R$  por  $\tilde{R}$ .

Ahora se obtendrá el vector de translación  $t$ , que describe la posición relativa de los orígenes de los dos marcos de referencia.

$$t = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \quad (5.20)$$

El vector  $t$  determina la posición en la que se encuentra el objeto respecto a la cámara. Partiendo de la ecuación (5.5) es posible calcularlo:

$$\begin{aligned}\tilde{X} &= K [R|t]\tilde{\Lambda} \\ K^{-1}\tilde{X} &= K^{-1}K [R|t]\tilde{\Lambda} \\ K^{-1}\tilde{X}\tilde{\Lambda}^{-1} &= [R|t]\tilde{\Lambda}\tilde{\Lambda}^{-1}\end{aligned}$$

$$[R|t] = K^{-1}\tilde{X}\tilde{\Lambda}^{-1} \quad (5.21)$$

$$\begin{aligned}&= \begin{pmatrix} \frac{1}{f} & 0 & \frac{-u_0}{f} \\ 0 & \frac{1}{f} & \frac{-v_0}{f} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{pmatrix} \begin{pmatrix} \frac{1}{l_1} & 0 & 0 & 0 \\ 0 & \frac{1}{l_2} & 0 & 0 \\ 0 & 0 & \frac{1}{l_3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \frac{x_{11}-u_0x_{31}}{fl_1} & \frac{x_{12}-u_0x_{32}}{fl_2} & \frac{x_{13}-u_0x_{33}}{fl_3} & \frac{x_{14}-u_0x_{34}}{f} \\ \frac{x_{21}-v_0x_{31}}{fl_1} & \frac{x_{22}-v_0x_{32}}{fl_2} & \frac{x_{23}-v_0x_{33}}{fl_3} & \frac{x_{24}-v_0x_{34}}{f} \\ \frac{x_{31}}{l_1} & \frac{x_{32}}{l_2} & \frac{x_{33}}{l_3} & x_{34} \end{pmatrix} \quad (5.22)\end{aligned}$$

El vector de translación  $t$  está dado en la última columna de la matriz  $[R|t]$ .

### Estimación de los ángulos de rotación

La matriz  $R$  puede ser definida por tres ángulos de Euler  $(\alpha, \beta, \gamma)$ . En esta tesis se trabajó de acuerdo a la relación dada por  $R = R_x(\alpha)R_y(\beta)R_z(\gamma)$ , donde cada una de las matrices que definen la rotación en cada eje, así como la matriz  $R$  resultante se muestra a continuación.

Matriz de rotación sobre el eje z:

$$R_z = \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Matriz de rotación sobre el eje y:

$$R_y = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix}$$

Matriz de rotación sobre el eje x:

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix}$$

Luego la matriz de rotación  $R = R_x(\alpha)R_y(\beta)R_z(\gamma)$ :

$$R = \begin{pmatrix} \cos \beta \cos \gamma & -\cos \beta \sin \gamma & \sin \beta \\ \sin \alpha \sin \beta \cos \gamma + \cos \alpha \sin \gamma & -\sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & -\sin \alpha \cos \beta \\ -\cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma & \cos \alpha \sin \beta \sin \gamma + \sin \alpha \cos \gamma & \cos \alpha \cos \beta \end{pmatrix}$$

Para obtener los tres ángulos  $(\alpha, \beta, \gamma)$ , igualamos esta matriz con la matriz de rotación que ya conocemos. Finalmente los ángulos están dados en base a las realaciones siguientes:

▪  $\beta$

$$r_{13} = \sin \beta$$

$$\beta = \sin^{-1}(r_{13}) \tag{5.23}$$

▪  $\alpha$

$$\begin{aligned} r_{23} = -\sin \alpha \cos \beta &\implies \cos \beta = \frac{-r_{23}}{\sin \alpha} \\ r_{33} = \cos \alpha \cos \beta &\implies \cos \beta = \frac{r_{33}}{\cos \alpha} \\ &= \frac{r_{33}}{\cos \alpha} \\ \frac{\sin \alpha}{\cos \alpha} = \frac{-r_{23}}{r_{33}} &\implies \tan \alpha = \frac{-r_{23}}{r_{33}} \end{aligned}$$

▪  $\gamma$ :

$$\begin{aligned} r_{11} = \cos \beta \cos \gamma &\implies \cos \beta = \frac{r_{11}}{\cos \gamma} \\ r_{12} = -\cos \beta \sin \gamma &\implies \cos \beta = \frac{-r_{12}}{\sin \gamma} \\ &= \frac{-r_{12}}{\sin \gamma} \\ \frac{\sin \gamma}{\cos \gamma} = \frac{-r_{12}}{r_{11}} &\implies \tan \gamma = \frac{-r_{12}}{r_{11}} \end{aligned}$$

### 5.3.4. Coordenadas del mundo y de la imagen

El sistema de coordenadas del mundo está definido de la siguiente forma:

$$P_w = \begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} \tag{5.24}$$

Las coordenadas homogéneas correspondientes para la ecuación anterior son:

$$P_w = \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \quad (5.25)$$

La asociación de los parámetros extrínsecos se realiza mediante una matriz aumentada en columnas, con el objeto de tener un sistema de coordenadas homogéneo en el mundo y relacionarlos así con el mundo de la cámara. La ecuación (5.26) muestra esta asociación:

$$P_c = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \quad (5.26)$$

Podemos ver que la ecuación (5.26) es análoga a la ecuación (5.1). Para recuperar cualquier posición del mundo con respecto de la cámara sólo es necesario calcular  $p \sim M P$ . Para este sistema de RA 3D, se utiliza el cálculo de puntos de la imagen de acuerdo a la parametrización de la base del paralelepípedo compuesta por las coordenadas  $(P_5, P_6, P_7, P_8)$ , y  $M$  que esta definida como  $M = K [R|t]$ .

## 5.4. Adición de información al marco

La tecnología de RA permite percibir con mayor realidad la composición de objetos generados por computadora en una escena real. Esta escena puede ser una secuencia de imágenes, procesamiento de video en tiempo real, ó técnicas de procesamiento de imágenes.

La descripción de objetos que se añadieron a la escena, así como la visualización de esta fase del sistema de RA 3D se explicarán completamente en esta sección.

### 5.4.1. Objetos virtuales

Un objeto virtual es un elemento digital, el cual es generado mediante técnicas de procesamiento digital de imágenes. El objeto virtual que se utilizó en este sistema de RA 3D es un círculo de color blanco, como el que se muestra en la figura 5.4. Éste objeto se moverá dentro del paralelepípedo (la representación del movimiento visual será respecto a la base del paralelepípedo).



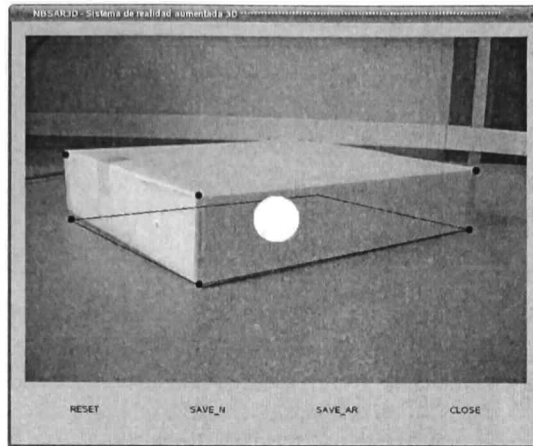


Figura 5.4: Marco de video con esquinas reconocidas, marcadas con círculos negros, y el objeto virtual dentro de la caja.

Otros objetos virtuales que se han añadido en el sistema de RA 3D, los cuales son: pequeños círculos en color negro que simulan las esquinas del objeto, y las líneas que delimitan la base del paralelepípedo y que han sido establecidas mediante la estimación de coordenadas de la imagen, estos objetos virtuales son opcionales y sólo sirven para verificar el movimiento del paralelepípedo. El movimiento del objeto virtual principal se explicará más adelante en este capítulo.

## 5.5. Descripción de la interfaz

La interfaz se compone de un widget creado en GTK y QT que muestra el procesamiento de video en tiempo real y cuatro botones con funciones específicas ( *RESET*, *SAVE\_N*, *SAVE\_AR*, *CLOSE* ), como se muestra en la figura 5.5. Las funciones que realizan estos botones se describen a continuación:

- \* *RESET*: Re-establece la búsqueda de marcas (esquinas) en el marco en ese momento.
- \* *SAVE\_N*: Guarda el marco que se está observando en formato *.pgn*; la imagen contiene únicamente la escena real.
- \* *SAVE\_AR*: Guarda el marco que se está observando en formato *.pgn*; la imagen contiene la escena real y todos los elementos virtuales que se tengan.

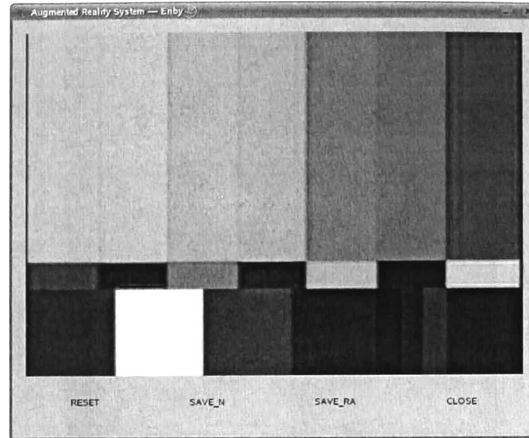


Figura 5.5: Interfaz del sistema.

- \* **CLOSE**: Finaliza todos los procesos del sistema y cierra el widget que muestra el contenido del video.

## 5.6. Movimiento físico del objeto virtual

El movimiento del objeto se realizó mediante la teoría de un *bloque sobre un plano inclinado*, al igual que el movimiento del sistema de RA 2D, sólo que para este sistema se realizan dos movimientos simultáneos en base a dos de los ángulos de rotación que se encontraron mediante la estimación de ángulos de rotación, en específico los ángulos  $\alpha$ , que define la inclinación en el eje  $x$ , y  $\beta$  que define la inclinación en el eje  $y$ .

Consideremos un *bloque<sub>1</sub>* de masa  $m$ , colocado sobre un plano inclinado que forma un ángulo  $\alpha$  con la horizontal, y otro *bloque<sub>2</sub>* de masa  $m$  colocado sobre un plano inclinado que forma un ángulo  $\beta$  con la horizontal, como se aprecia en la figura 5.6, donde también se muestran los diagramas de cuerpo libre de los bloques. Las fuerzas que actúan sobre ellos son: la normal ( $N$ ) que actúa perpendicular al plano, y el peso ( $mg$ ) que actúa verticalmente hacia abajo. Para darle un enfoque más real al movimiento del objeto virtual se añadió una constante de fricción.

Como se definió en el capítulo 4, en la sección del movimiento físico del sistema de RA 2D, al efecto producido cuando la superficie de un cuerpo se desliza sobre otro se le conoce como *fuerza de fricción*. La razón entre la magnitud de la fuerza máxima de fricción estática y la magnitud de la fuerza normal se llama *coeficiente de fricción estática* de las superficies implicadas. Si  $f_s$  representa la magnitud de la fuerza de

fricción estática, entonces se cumple que:

$$f_s \leq \mu_s N \quad (5.27)$$

donde  $\mu_s$  es el coeficiente de fricción estática y  $N$  es la magnitud de la fuerza normal. El signo de igualdad se tiene sólo cuando  $f_s$  alcance su valor máximo.

- Movimiento sobre el eje  $x$ .

$$\begin{aligned} \Sigma F_x &= mg \sin \alpha - f_s = ma_x \\ \Sigma F_y &= N - mg \cos \alpha = 0 \end{aligned}$$

Por definición tenemos que  $f_s = \mu_s N$  y  $N = mg \cos \alpha$ . Despejando  $N$  de la suma de fuerzas en  $y$ , obtenemos que:

$$f_s = \mu_s mg \cos \alpha \quad (5.28)$$

La aceleración en el eje  $x$  es :

$$\begin{aligned} a_x &= \frac{(mg \sin \alpha - \mu_s mg \cos \alpha)}{m} \\ &= \frac{mg(\sin \alpha - \mu_s \cos \alpha)}{m} \\ a_x &= g(\sin \alpha - \mu_s \cos \alpha) \end{aligned} \quad (5.29)$$

Es importante mencionar que la ecuación (5.29) está calculada en base a una inclinación positiva definida por el eje  $x$  que se ha manejado en el paralelepípedo a lo largo del desarrollo de este sistema de RA 3D. Cuando hay cambio de inclinación en el paralelepípedo, entonces la aceleración en  $x$  esta dada por la ecuación

$$a_x = g(\mu_s \cos \alpha - \sin \alpha) \quad (5.30)$$

- Movimiento sobre el eje  $y$ .

La definición de la aceleración que se da en el eje  $y$  es análoga a la estimación de la aceleración en el eje  $x$ , su ecuación de aceleración resultante es:

$$a_y = g(\mu_s \cos \beta - \sin \beta) \quad (5.31)$$

ó bien

$$a_y = g(\sin \beta - \mu_s \cos \beta) \quad (5.32)$$

La ecuación (5.31) se da cuando la inclinación es positiva, ya que se tendrá que disminuir la coordenada de  $y$  en el marco, cuando se da el caso contrario, entonces se utilizará la aceleración  $a_y$  dada por la ecuación (5.32).

Para ambos movimientos en los ejes  $x, y$  se concluye que:

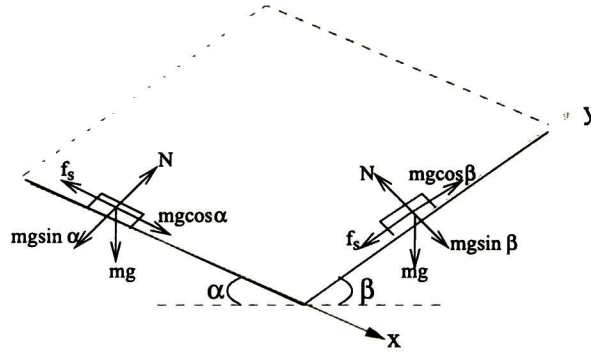


Figura 5.6: Movimiento físico del objeto virtual en el sistema de realidad aumentada 3D.

- \* La aceleración a lo largo del plano inclinado está dada por la componente de peso a lo largo del plano:  $a_x = g(\sin \alpha - f_s)$ ,  $a_y = g(f_s - \sin \beta)$ .
- \* La componente del peso perpendicular al plano esta equilibrada por la fuerza normal tal que  $N = mg \cos \alpha$ ,  $= mg \cos \beta$ .
- \* La aceleración  $a_x, a_y$  es independiente de la masa del bloque, únicamente depende del ángulo ( $\alpha$  ó  $\beta$ ) y de  $g$ .
- \* La aceleración es constante.

Como las aceleraciones en cada eje son constantes, se utilizaron las ecuaciones para velocidad y desplazamiento siguientes:

Ecuaciones que definen la velocidad en un momento  $t$ .

$$v_x = v_{x0} + a_x t \quad (5.33)$$

$$v_y = v_{y0} + a_y t \quad (5.34)$$

Ecuaciones que definen el desplazamiento del objeto en un momento  $t$ .

$$x = x_0 + v_x t + \frac{1}{2} a_x t^2 \quad (5.35)$$

$$y = y_0 + v_y t + \frac{1}{2} a_y t^2 \quad (5.36)$$

En cada marco se calcularán los ángulos de rotación del paralelepípedo. En base a esto y conforme a las ecuaciones definidas anteriormente se determinará si el objeto

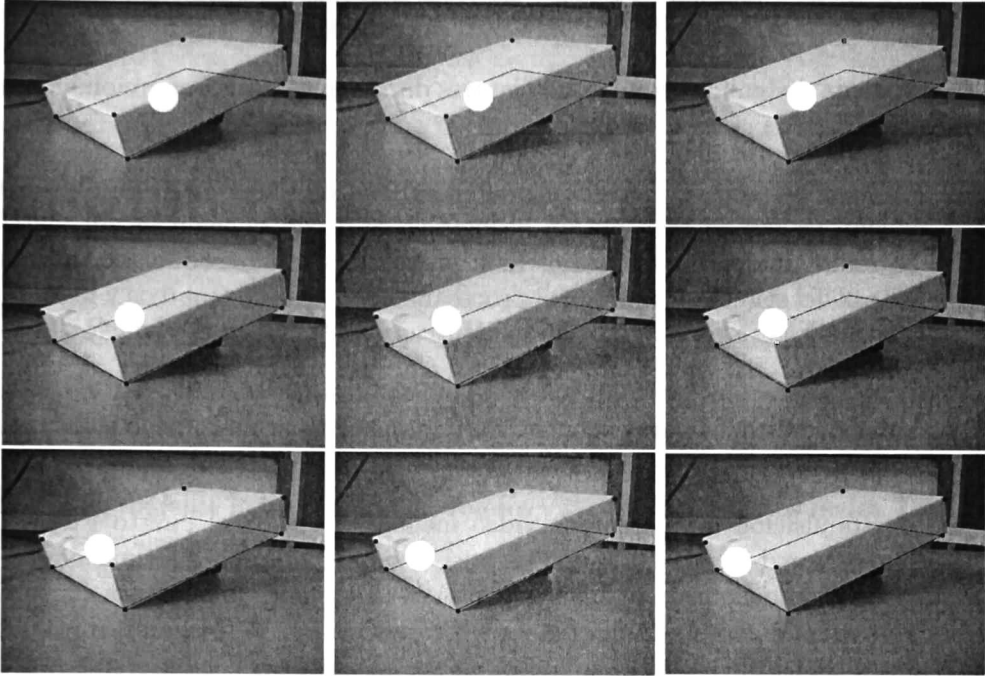


Figura 5.7: Secuencia de movimiento del objeto virtual en el sistema de RA 3D.

debe moverse o no. Para calcular la nueva posición del objeto virtual se actualizarán las coordenadas paramétricas del objeto con respecto al desplazamiento producido por las ecuaciones [5.35, 5.36] ( $m_x = m_x + x, m_y = m_y + y$ ), después estas coordenadas se mandan a una función que calcula la nueva posición en coordenadas de la imagen, como se mostró en la ecuación (5.1), es posible obtener un punto de la imagen, aplicando los parámetros extrínsecos representados por  $M = K \cdot [R|t]$ , únicamente ahora se define  $P$  como  $(m_x, m_y, 0, 1)$ . Esta representación se puede observar en la relación siguiente:

$$p \sim M \begin{pmatrix} m_x \\ m_y \\ 0 \\ 1 \end{pmatrix} \tag{5.37}$$

En la figura 5.7 se muestra una secuencia de imágenes en las que se aprecia el movimiento del objeto virtual en el paralelepípedo para 9 marcos. Los ángulos de rotación detectados en esta secuencia de marcos están en el siguiente promedio; los valores se expresan en grados:  $\alpha = -49.136^\circ$   $\beta = 10.268^\circ$ .

## 5.7 Resultados

El sistema de realidad aumentada en tres dimensiones nos proporcionó los siguientes resultados:

- \* En este sistema de RA 3D el procesamiento de video en tiempo real es muy bueno, ya que la transmisión de marcos se da en relación de un marco cada 30 milisegundos en promedio. Este proceso no se ve modificado por las otras etapas del sistema, como son el procesamiento digital de marcos y los algoritmos utilizados para obtener las características geométricas del paralelepípedo.
- \* La segmentación por color que se le aplica a este sistema de RA 3D, nos ofrece buenos resultados para encontrar las esquinas del paralelepípedo en comparación de otras técnicas que suelen ser más costosas computacionalmente.
- \* La adquisición de marcos de video mediante el uso del puerto Firewire y las bibliotecas de Kino, GTK/QT es bueno; sin embargo, la información del marco viene con pequeñas alteraciones (ruido en la imagen), lo que hace que el sistema no sea totalmente estable. Esto provoca que se den desplazamientos (mínimos) entre las coordenadas de una esquina en un marco  $n$  y un marco  $n + 1$ .
- \* La interacción entre el objeto virtual y el paralelepípedo nos permite simular sucesos físicos de una manera totalmente práctica y correcta.

# Capítulo 6

## Conclusiones y trabajo futuro

Los sistemas de realidad aumentada que se desarrollaron en este trabajo de tesis muestran claramente la interacción de objetos virtuales con objetos reales. Las etapas que se desarrollaron en esta tesis fueron:

- Procesamiento de video.
- Procesamiento digital de marcos de video.  
Sistema de realidad aumentada en dos dimensiones.
- Sistema de realidad aumentada en tres dimensiones.

Cada una de estas etapas se cumplió eficazmente y sus conclusiones se mencionan a continuación.

### 6.1. Etapa de procesamiento de video

El procesamiento de video se realizó utilizando bibliotecas de Kino y de GTK/QT, la transmisión de video se dio en un promedio de 25 marcos por segundo, la calidad del video es alta; sin embargo, nos dimos cuenta que el video recibido por el puerto Firewire, mediante las bibliotecas antes mencionadas, contiene ruido. Esto implica que se vea un pequeño movimiento externo (mínimo) en la visualización de los sistemas de realidad aumentada. Sin embargo la transmisión de video es muy eficiente, por lo que el funcionamiento de todo el despliegado de los sistemas de realidad aumentada es muy bueno.

Es importante mencionar que la extracción de marcos (imágenes) de video que se requieren en la etapa de *procesamiento digital de marcos* se extraen directamente de memoria, por lo que los recursos de memoria de la computadora no se ven afectados.

## 6.2. Etapa de procesamiento digital de marcos

En esta etapa se utilizaron varias técnicas para mejorar la calidad del marco de video, como algoritmos de suavizado, segmentación por diferencia entre marcos, segmentación por color, detección de esquinas, y algunas operaciones morfológicas. Todo este procesamiento fue de vital importancia para la detección de parámetros que definieron la obtención de características geométricas del objeto real.

El procesamiento digital de marcos entrega muy buenos resultados, con un tiempo mínimo de procesamiento, esto implica que la visualización del video no se ve retrasada en ningún momento.

## 6.3. Sistema de realidad aumentada en dos dimensiones

En este sistema de realidad aumentada la interacción entre el objeto virtual y el objeto real, se visualiza de una forma totalmente natural.

Gracias al correcto funcionamiento de detección de esquinas, al seguimiento de marcas, a la obtención de características geométricas, y la especificación de movimiento virtual a través del sistema físico basado en el *movimiento de un bloque sobre un plano inclinado*, es posible ver el deslizamiento del objeto virtual (círculo), a través del objeto real (rectángulo), de tal forma que el objeto virtual parece formar parte de la escena real, además se añadió una componente de fricción para reducir la velocidad en el deslizamiento del objeto virtual, proporcionándole más realismo al movimiento.

## 6.4. Sistema de realidad aumentada en tres dimensiones

El resultado en el movimiento del objeto virtual es de un alto grado de similitud con respecto al movimiento físico de los objetos que interactúan en el mundo real, tenemos que tomar en cuenta que este movimiento esta definido por la estimación de componentes geométricas que establecen la orientación e inclinación del objeto real, en donde esta estimación se ve afectada por el ruido que se tiene en el marco adquirido por el puerto Firewire y la ayuda de bibliotecas de GTK/Qt y Kino.

Es importante mencionar que el sistema de realidad aumentada en tres dimensiones se ha especificado para una vista en perspectiva, por lo que se encuentra restringido a una perspectiva máxima y mínima, en la que se se pueden calcular los componentes necesarios para definir el cuerpo geométrico del objeto, sin embargo el movimiento



del objeto virtual se aprecia perfectamente en las diferentes inclinaciones posibles del paralelepípedo.

En ambos sistemas se realiza todo el proceso de manera automática, y sin la intervención del usuario para establecer características que pudieran facilitar la interacción del objeto virtual con la escena real, sólo son necesarias algunas marcas para definir esquinas de los objetos reales.

## **6.5. Trabajo futuro**

Con la finalidad de tener una herramienta robusta que pueda ser utilizada para diferentes problemas, es necesario optimizar algunos aspectos de los sistemas de realidad aumentada que se desarrollaron.

Un aspecto importante en el que se deberá trabajar es la estabilización del sistema, para esto es necesario investigar algunas otras opciones que nos permitan obtener video desde el puerto sin ningún tipo de compresión del video.

Otro punto importante es la migración del sistema a un ambiente tridimensional, como puede ser OpenGL, con la finalidad de que los objetos virtuales que aparezcan en la escena tengan profundidad y dimensión, esto tendrá un efecto más realista de la realidad en la escena.

Una característica importante para la interacción de los objetos virtuales y los objetos reales, es la aplicación de métodos que establezcan la oclusión del objeto virtual con respecto al objeto real, definiendo restricciones que delimiten la visualización del objeto virtual.

Una mejora importante para este trabajo de tesis, será incluir algoritmos que obtengan eficientemente las esquinas de los objetos reales, para prescindir del marcado físico.

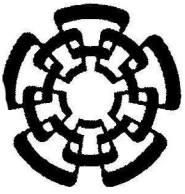
Finalmente, se pretende que este sistema tenga una colección de simulaciones con diferentes tipos de objetos virtuales y objetos reales, por lo que se desea que poco a poco se vayan incorporando nuevas opciones de elementos interactivos entre la escena real y la adición de información virtual.

# Bibliografía

- [1] Canon Camcorders Features GL2, April 2006. <http://consumer.usa.canon.com>.
- [2] IEEE 1394, April 2006. [http://es.wikipedia.org/wiki/IEEE\\_1394](http://es.wikipedia.org/wiki/IEEE_1394).
- [3] Video Digital, 2006. <http://redescolar.ilce.edu.mx/redescolar/cursos/>.
- [4] Ronald T. Azuma. Augmented reality: Approaches and technical challenges. In Woodrow Barfield and Thomas Caudell, editors, *In Fundamentals of Wearable Computers and Augmented Reality*, pages 27–63. Lawrence Erlbaum Associates, 2001.
- [5] Manfred Bogen. *ARSys-Tricoder, AR System in Medicine (transplant design)*. Department of Virtual Environments, Fraunhofer-Institute for Media Communication IMK, D-53754 Sankt Augustin, 2003. <http://www.arsys-tricorder.org/>.
- [6] Bruno Caprile and Vincent Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision.*, 4(2):127–140, 1990.
- [7] R. Cipolla, T. Drummond, and D. Robertson. Camera calibration from vanishing points in images of architectural scenes. *BMVC99*, page 10, 1999.
- [8] Farbiz F., Cheok A.D., Zhou ZhiYing Liu Wei, Xu Ke, Prince S., Billingham M., and Kato H. Live three-dimensional content for augmented reality. *IEEE Transactions on Multimedia.*, 7(3):514–523, 2005.
- [9] Adrian Ford and Alan Roberts. *Colour Space Conversions*, July 2001. <http://www.poynton.com/PDFs/coloureq.pdf>.
- [10] Bruce Fraser. *Color*. Adobe Magazine, San Francisco, November 1996. <http://www.adobe.com/products/adobemag/archive/pdfs/9611febf.pdf>.
- [11] The GNOME Project. *GdkPixbuf.*, 2003 2004. <http://developer.gnome.org/arch/imaging/gdkpixbuf.html>.
- [12] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

- [13] GTK+ Reference Manual. *GtkWidget*.  
<http://developer.gnome.org/doc/API/2.0/gtk/GtkWidget.html> .
- [14] Nikos Ioannidis. *MEDARPA, MEDical Augmented Reality for PATients*. ZGDV - Zentrum für Graphische Datenverarbeitung, Fraunhoferstr. 5, 64283 Darmstadt.  
<http://www.medarpa.de/>.
- [15] Nikos Ioannidis. *ARCHEOGUIDE, Augmented Reality-based cultural Heritage On-site Guide*. Development Programmes Department, 19002 Peania, Greece, 2002. <http://archeoguide.intranet.gr/>.
- [16] David Jelinek and Camillo J. Taylor. Reconstruction of linearly parameterized models from single images with a camera of unknown focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(7):7, 2001.
- [17] Jim Vallino, Department of Software Engineering, Rochester Institute of Technology, 134 Lomb Memorial Drive, Rochester, NY 14623-5608. *Introduction to Augmented Reality*, 2002. <http://www.se.rit.edu/~jrv/research/ar/>.
- [18] Hyun-Kook Kahng, editor. *Information Networking, Networking Technologies for Enhanced Internet Services International Conference, ICOIN 2003.*, volume 2662 of *Lecture Notes in Computer Science*. Springer, 2003.
- [19] Andrei Lintu. An augmented reality system for astronomical observations. In *IEEE Virtual Reality 2006*, pages 119–126, Alexandria, Virginia, USA, 2006. IEEE Computer Society, IEEE.
- [20] Wilczkowiak Marta, Boyer Edmond, and Sturm Peter. Camera calibration and 3d reconstruction from single images using parallelepipeds. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, volume 1, pages 142–148. IEEE Computer Society Press, Jul 2001.
- [21] Alex Olwal, Christoffer Lindfors, Jonny Gustafsson, Torsten Kjellberg, and Lars Mattsson. Astor: An autostereoscopic optical see-through augmented reality system. In *ISMAR*, pages 24–27, 2005.
- [22] Open Source Organization. *Linux DV, Digital Video for Linux*, April 2006. <http://www.kinodv.org/>.
- [23] Wayne Piekarski. 3d modeling with the tinmith mobile outdoor augmented reality system. *IEEE Comput. Graph. Appl.*, 26(1):14–17, 2006.
- [24] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C++: The Art of Scientific Computing*. Press Syndicate of the University of Cambridge, 2002.

- [25] Serway R. *Física Vol. 1*. McGraw-Hill, 1992.
- [26] Resnick Robert, Halliday David, and Krane Kenneth S. *Física Vol. 1*. Continental de México, 1992.
- [27] De Buck S., F. Maes, J. Ector, Bogaert J., Dymarkowski S., Heidbuchel H., and Suetens P. An augmented reality system for patient-specific guidance of cardiac catheter ablation procedures. In *IEEE Transactions on Medical Imaging*, pages 1512–1524. IEEE, IEEE, 2005.
- [28] Fei Shen and Han Wang. Corner detection based on modified Hough transform. *Pattern Recogn. Lett.*, 23(8):1039–1049, 2002.
- [29] P. Smith, D. Sinclair, R. Cipolla, and K. Wood. Effective corner matching, 1998.
- [30] S. M. Smith and J. M. Brady. SUSAN – A new approach to low level image processing. Technical Report TR95SMS1c, Chertsey, Surrey, UK, 1995.
- [31] Emanuele Trucco and Alessandro Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.
- [32] S. White, Feiner S., and Kopylec J. Virtual vouchers: Prototyping a mobile augmented reality user interface for botanical species identification. In *IEEE Virtual Reality 2006*, pages 133–133. IEEE Computer Society, IEEE, 2006.
- [33] Wolfram MathWorld. *Vanishing Point*, 1999 2006. <http://mathworld.wolfram.com/VanishingPoint.html>.
- [34] Zhengyou Zhang. Camera calibration with one-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 26(7):892–899, 2004.



# CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N. UNIDAD GUADALAJARA

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Sistema de Realidad Aumentada

del (la) C.

Norma Irene SERNA RODRÍGUEZ

el día 16 de Noviembre de 2006.

Dr. Raúl Ernesto González Torres  
Investigador CINVESTAV 2C  
CINVESTAV Unidad Guadalajara

Dr. Félix Francisco Ramos Corchado  
Investigador CINVESTAV 2B  
CINVESTAV Unidad Guadalajara

Dr. Luis Gerardo de la Fraga  
Jefe de la Sección de Computación  
CINVESTAV México

Dr. Mario Angel Siller González  
Pico  
Investigador CINVESTAV  
CINVESTAV Unidad Guadalajara



CINVESTAV  
BIBLIOTECA CENTRAL



SSIT000006613