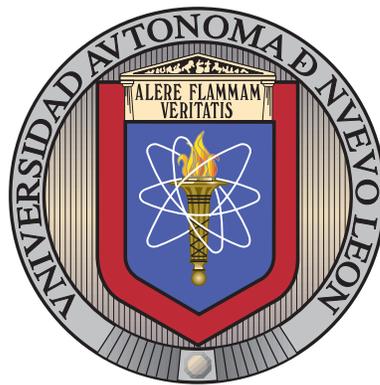


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE LICENCIATURA



USO DE MARCADORES PARA LA SUPERPOSICIÓN
DE MODELOS TRIDIMENSIONALES EN IMÁGENES

POR

EMMANUEL ALEJANDRO GARCÍA SOLÍS

EN OPCIÓN AL GRADO DE

INGENIERO EN TECNOLOGÍA DE SOFTWARE

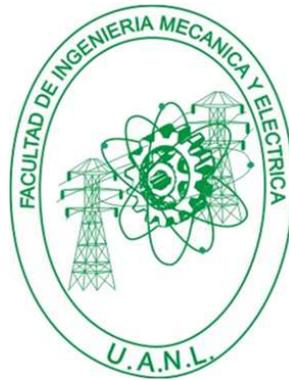
SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

NOVIEMBRE 2014

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE LICENCIATURA



USO DE MARCADORES PARA LA SUPERPOSICIÓN
DE MODELOS TRIDIMENSIONALES EN IMÁGENES

POR

EMMANUEL ALEJANDRO GARCÍA SOLÍS

EN OPCIÓN AL GRADO DE

INGENIERO EN TECNOLOGÍA DE SOFTWARE

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

NOVIEMBRE 2014

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

División de Estudios de Licenciatura

Los miembros del Comité de Tesis recomendamos que la Tesis «Uso de marcadores para la superposición de modelos tridimensionales en imágenes», realizada por el alumno Emmanuel Alejandro García Solís, con número de matrícula 1450138, sea aceptada para su defensa como opción al grado de Ingeniero en Tecnología de Software.

El Comité de Tesis

Dra. Satu Elisa Schaeffer

Asesor

Dra. Iris Abril Martínez Salazar

Revisor

M.C. Carlos Bernardo Garza Treviño

Revisor

M.C. Arnulfo Treviño Cubero

División de Estudios de Licenciatura

San Nicolás de los Garza, Nuevo León, noviembre 2014

AGRADECIMIENTOS

Quiero extender agradecimientos primeramente a mi familia por el apoyo recibido durante todos mis estudios. Sin ellos no me habría sido posible haber llegado hasta aquí.

También quiero agradecer a la Universidad Autónoma de Nuevo León y especialmente a la Facultad de Ingeniería Mecánica y Eléctrica por mi estancia durante mis estudios de licenciatura en la carrera de Ingeniería en Tecnología de Software.

Además quiero agradecer al comité de tesis integrado por la Dra. Satu Elisa Schaeffer como asesora, Dra. Iris Abril Martínez Salazar y M.C. Carlos Bernardo Garza Treviño como revisores.

Agradecimientos especiales a la Dra. Elisa Schaeffer por la orientación proporcionada durante una gran parte de mi carrera y por su ayuda brindada en la realización de esta tesis.

RESUMEN

Emmanuel Alejandro García Solís.

Candidato para el grado de Ingeniero en Tecnología de Software.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio:

USO DE MARCADORES PARA LA SUPERPOSICIÓN DE MODELOS TRIDIMENSIONALES EN IMÁGENES

Número de páginas: 52.

OBJETIVO: El objetivo de este trabajo es la implementación de un sistema de realidad aumentada habilitado mediante la detección de marcadores para el consecuente estudio de las ventajas, desventajas y aplicaciones que pueden existir para este tipo de sistemas.

MÉTODO: Análisis, diseño e implementación de un sistema de realidad aumentada. El primer paso, análisis, refiere a toda la recopilación de información relacionada, referencias de estudio, selección de herramientas, calendarización de etapas del proyecto y demás consideraciones necesarias a conocer previo al desarrollo de un sistema computacional.

En la etapa de diseño se plantean ideas iniciales para establecer una arquitectura a seguir más adelante en el desarrollo del sistema. Se definen elementos como la comunicación entre las partes del sistema, la separación del código de manera que sea eficiente y fácil de comprender, entre otras consideraciones importantes a tomar de tal forma que al momento de iniciar el desarrollo del sistema quede clara la forma en la que éste se implementará.

Finalmente, en la etapa de implementación se realiza la tarea de la codificación del sistema, utilizando todas las consideraciones anteriores (herramientas, referencias, arquitectura, etcétera).

CONTRIBUCIONES: La contribución de este trabajo corresponde a un sistema que permite la detección de un cierto marcador en una imagen o secuencia de imágenes para lograr la superposición de un modelo tridimensional en la posición del marcador con respecto a la imagen.

Dicho modelo tridimensional puede variar, siendo posible asignar uno diferente por cada marcador existente en una imagen. Además, se estudia el comportamiento de este tipo de sistemas comparando un estado por defecto, es decir, usando un solo marcador, con su desempeño cuando se utilizan múltiples marcadores.

Firma del asesor: _____

Dra. Satu Elisa Schaeffer

ÍNDICE GENERAL

1. Introducción	1
1.1. Hipótesis y objetivos	1
1.2. Estructura de la tesis	2
2. Antecedentes	4
2.1. Realidad virtual	4
2.2. Realidad aumentada	6
2.3. Visión computacional	7
2.4. Códigos QR	11
3. Literatura existente	14
3.1. Trabajos similares	14
3.2. Clasificación de trabajos similares	16
4. Solución propuesta	18
4.1. Metodología	18

4.1.1. Análisis	19
4.1.2. Diseño	22
4.2. Marco teórico	24
4.2.1. Visión computacional	24
4.2.2. Programación tridimensional	28
4.3. Implementación	30
4.3.1. Algoritmo	30
4.3.2. Procesamiento de imágenes	32
5. Experimentos	36
5.1. Un solo código QR	37
5.2. Múltiples códigos QR	41
6. Conclusiones	45
6.1. Contribuciones	45
6.2. Trabajo a futuro	46
Bibliografía	48

ÍNDICE DE FIGURAS

2.1. Oculus Rift, hardware de realidad virtual.	5
2.2. Diferencia entre realidad aumentada y realidad virtual.	6
2.3. Aplicaciones de la visión computacional.	8
2.4. Procesamiento de una imagen.	10
2.5. Ejemplos de códigos QR.	12
2.6. Código QR detectado y decodificado.	13
4.1. Diagrama de la metodología utilizada.	19
4.2. Arquitectura del sistema propuesto.	23
4.3. Diagrama de clases del sistema.	24
4.4. Interpolación lineal.	25
4.5. Interpolación bilineal.	26
4.6. Método Otsu para la binarización de una imagen.	28
4.7. Cubo descrito en el formato Wavefront OBJ.	30
4.8. Cubo creado a partir de un archivo Wavefront OBJ.	30

4.9. Diagrama de flujo del sistema propuesto.	31
4.10. Extracción de una imagen de un video.	33
4.11. Conversión a escala de grises y binarización de la imagen extraída. . .	33
4.12. Código QR detectado y decodificado.	34
4.13. Modelo superpuesto en el código detectado.	35
5.1. Modelo 3D de un cubo con un solo código QR.	38
5.2. Modelo 3D de una rueda con un solo código QR.	39
5.3. Modelo 3D de una silla con un solo código QR.	40
5.4. Superposición de dos modelos simultáneos.	42
5.5. Superposición de seis modelos simultáneos.	43

ÍNDICE DE CUADROS

3.1. Clasificación de trabajos similares.	17
4.1. Herramientas de hardware y software utilizadas.	21
4.2. Selección de modelos tridimensionales.	35

CAPÍTULO 1

INTRODUCCIÓN

A continuación se presenta una descripción breve de lo que se busca solucionar con este trabajo, la motivación que lleva a la realización de la tesis, la hipótesis y objetivos planteados y la estructura que seguirán los capítulos posteriores.

MOTIVACIÓN

La realidad aumentada se está convirtiendo en una tecnología común hoy en día. Con la llegada de los teléfonos inteligentes y las tabletas, la realidad aumentada se convertirá en el acercamiento adecuado para aplicaciones que interactúan con la cámara de dichos dispositivos. Debido a ello se decidió a llevar a cabo la implementación de un sistema de realidad aumentada.

1.1 HIPÓTESIS Y OBJETIVOS

En esta sección se plantea la hipótesis a evaluar dentro de este trabajo de tesis, así como los objetivos que se definen para el proyecto.

HIPÓTESIS

La implementación de un sistema de realidad aumentada mediante la detección de

marcadores habilitará la opción de la visualización de modelos tridimensionales sobrepuestos en imágenes simples, permitiendo la implementación de una gran variedad de aplicaciones.

OBJETIVOS

El objetivo principal de este trabajo es la implementación de un sistema de realidad aumentada utilizando marcadores en el campo de visión de la cámara de tal manera que sea posible superponer un modelo tridimensional sobre una imagen capturada desde la cámara.

Los *objetivos específicos* son:

- Implementación de un sistema que utiliza visión computacional para detectar marcadores en una imagen y superpone modelos tridimensionales sobre ellos.
- Evaluación del comportamiento de dicho sistema cuando se utilizan múltiples marcadores.
- Estudio del desempeño del sistema cuando se utilizan modelos tridimensionales complejos.

1.2 ESTRUCTURA DE LA TESIS

En esta sección se describe la estructura de la tesis, resumiendo brevemente el contenido que se verá en cada uno de los capítulos posteriores.

En el capítulo 2 se describen los antecedentes al tema de investigación de este trabajo. En este capítulo se discuten temas relacionados de cierta manera a la realidad aumentada, explicando en cada caso su relación y en que afecta al desarrollo de esta investigación.

El capítulo 3 presenta un listado de literatura existente (trabajos similares, artículos de revistas, páginas web, libros, etcétera), que guardan alguna relación con esta tesis. Además se realiza una comparación objetiva de cada una, clasificando cada trabajo en ciertas categorías específicas que tienen en común con la presente tesis.

En seguida en el capítulo 4 se explica a detalle la metodología usada en el desarrollo de la presente investigación, las etapas planteadas, y otros procesos considerados previo a la implementación de la solución. Además se explica detalladamente la implementación del sistema que se propone en este trabajo, mencionando paso a paso el funcionamiento del mismo y la teoría detrás de ello.

Después en el capítulo 5 se realizan diversos experimentos aplicados a la solución propuesta con el objetivo de estudiar el comportamiento del sistema al usarse con diversos parámetros diferentes a lo que se establece por defecto.

En el capítulo 6 se encuentran los comentarios finales del proyecto, además de una mención de las contribuciones que se realizaron con esta tesis y el posible trabajo a futuro a realizar.

CAPÍTULO 2

ANTECEDENTES

A continuación se presentan diversos campos o áreas relacionadas con la realidad aumentada que permiten la implementación de este tipo de sistemas. Conocer dichos campos es necesario para el correcto entendimiento de los temas que se mencionan más adelante en este trabajo, como son la realidad aumentada, tema principal de este trabajo, la realidad virtual, que se encuentra relacionada a la anterior, la visión computacional y las técnicas de procesamiento de imágenes usadas en la implementación del sistema, así como una descripción de los códigos *QR*, sus características y aplicaciones.

2.1 REALIDAD VIRTUAL

La *realidad virtual* [7] es la simulación de entornos y mecanismos sensoriales del hombre por medio de una computadora, de tal manera que se busca proporcionar al usuario la sensación de inmersión y la capacidad de interacción con ambientes artificiales. La realidad virtual puede ser: inmersiva y no inmersiva. La realidad virtual inmersiva con frecuencia se liga a un ambiente tridimensional creado por computadora, el cual se manipula a través de cascos, guantes u otros dispositivos que capturan la posición y rotación de diferentes partes del cuerpo humano. En cambio, la realidad virtual no inmersiva es aquella que se crea cuando el participante explora

diversos ambientes haciendo uso de los dispositivos de *hardware*¹ comunes: ratón, pantalla, tarjeta de sonido y bocinas.

Lo anterior define lo que se conoce como un ambiente virtual. Un ambiente virtual es una simulación por computadora que proporciona información a uno o varios de los sentidos de una persona sentidos: visión, sonido, tacto y gusto, con el propósito de que el usuario se sienta inmerso en un mundo que reacciona ante sus acciones.

Los ambientes virtuales son naturalmente tridimensionales, dinámicos y cambiantes. El usuario puede explorar y experimentar de acuerdo con las situaciones generadas como combinación de su interacción con el mundo virtual y la retroalimentación que éste proporciona.

Un ejemplo de realidad virtual es el Oculus Rift [12], un popular dispositivo de realidad virtual habilitado mediante un *HMD*² se puede observar en la figura 2.1.



(a) Hardware del Oculus Rift^a.



(b) Oculus Rift en acción^b.

Figura 2.1 – Oculus Rift, hardware de realidad virtual.

^aFuente: <http://www.mojandroid.sk/wp-content/uploads/2014/03/Oculus.jpg>

^bFuente: <http://assets1.ignimgs.com/2014/03/03/riftjpg-df0fcd.1280w.jpg>

¹Partes tangibles de un sistema informático.

²Del inglés *Head Mounted Display*, es un dispositivo de visualización montable en la cabeza que permite reproducir imágenes a través de una pantalla.

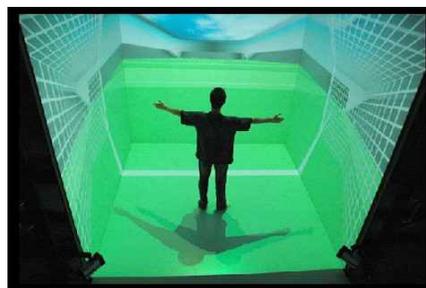
2.2 REALIDAD AUMENTADA

La realidad aumentada es una variación de realidad virtual. Las tecnologías de realidad virtual sumergen al usuario dentro de un entorno completamente sintético, sin tener consciencia del mundo real que lo rodea. La realidad aumentada, sin embargo, permite al usuario ver el mundo real, en el que se superponen o con el que se componen objetos virtuales. Así, la realidad aumentada no sustituye la realidad, sino que la complementa.

En la figura 2.2 se muestran ejemplos para mostrar la diferencia entre un sistema de realidad aumentada y uno de realidad virtual.



(a) Ejemplo de realidad aumentada^a.



(b) Ejemplo de realidad virtual^b.

Figura 2.2 – Diferencia entre realidad aumentada y realidad virtual.

^aFuente: <http://danmayes.files.wordpress.com/2012/09/augmented-reality-001.jpg>

^bFuente: <http://www.21stcentech.com/wp-content/uploads/2012/07/Virtual-Reality.jpg>

La realidad aumentada ha experimentado un creciente interés en estos últimos años. En septiembre de 2009, *The Economist* [4] aseguró que “intentar imaginar cómo se utilizará la realidad aumentada es como intentar predecir el futuro de la web en 1994”. Según la consultora *Juniper Research*, el mercado de la realidad aumentada generará más de 6000 millones de dólares en 2017 [15].

2.3 VISIÓN COMPUTACIONAL

La *visión computacional* es el estudio del procesamiento de las imágenes de tal manera que sea posible reconocer y localizar objetos en el ambiente [23].

La visión computacional es importante en los sistemas de realidad aumentada debido a que permite la detección de marcadores o patrones específicos sobre los cuales se busca superponer información virtual, siendo el objetivo de este trabajo la superposición de modelos tridimensionales en imágenes a partir de la detección de marcadores.

Ejemplos de aplicaciones de visión computacional son:

- Navegación autónoma [6].
- Conteo de personas que ingresan a un establecimiento [9].
- Organización y clasificación de imágenes [1].
- Detección de patrones en imágenes [3].

En la figura 2.3 se pueden observar ejemplos de los diferentes tipos de aplicaciones de la visión computacional.

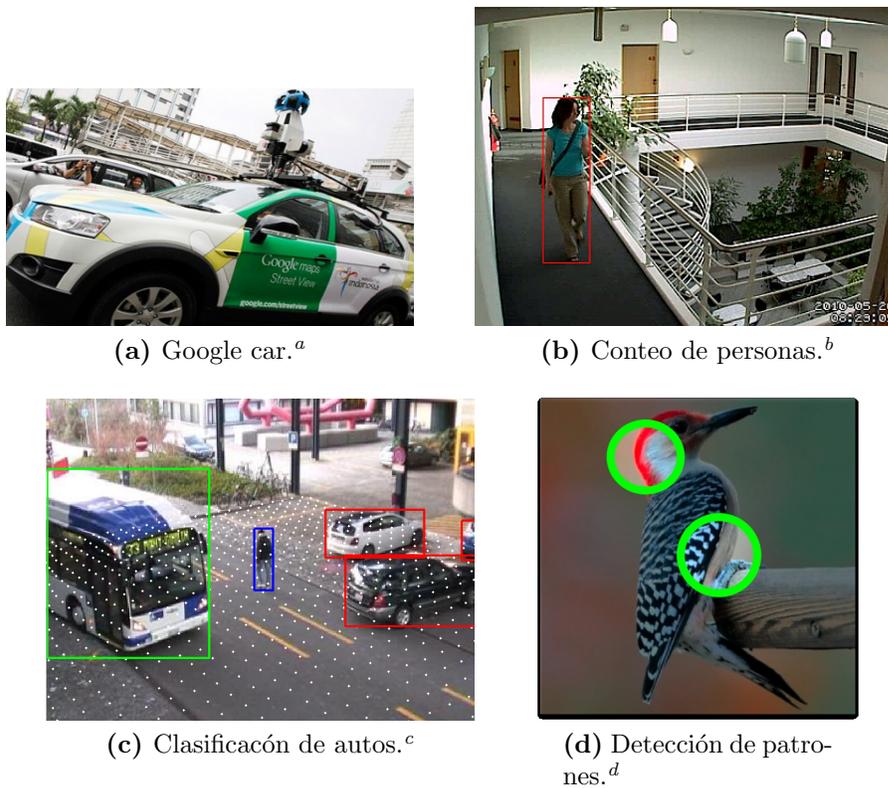


Figura 2.3 – Aplicaciones de la visión computacional.

^aFuente: http://vs01.frecuencialatina.com/videos/90/2014/04/29/7d43473439656ef5_big.jpg

^bFuente: <http://www.c-mor.com/img/camera-motion-detection-400.jpg>

^cFuente: <http://cvlab.epfl.ch/files/content/sites/cvlab2/files/data/multiclass/c2.png>

^dFuente: http://web.eecs.umich.edu/~jiadeng/img/bubble_icon.jpg

PROCESAMIENTO DE IMÁGENES

Un área muy relacionada a la de visión computacional es la de procesamiento de imágenes. El objetivo del procesamiento de imágenes es mejorar la calidad de las imágenes para su uso posterior, por ejemplo:

- Remover defectos.
- Remover problemas por movimiento o desenfoque.
- Mejorar ciertas propiedades como color, contraste, etc.

Tratamientos comunes aplicados a imágenes que se usan en la presente tesis involucran: conversión a escala de grises, *binarización*³ y reducción de ruido (suavizado).

El primer caso es la conversión a escala de grises. Los valores de los píxeles de una imagen normalmente se encuentran en formato *RGB*⁴ por lo que las técnicas para lograr esto generalmente involucran calcular un promedio entre los tres valores, o usar ya sea el menor, el mayor el valor de enmedio y repetirlo para los tres valores. Esto convierte una imagen a escala de grises, como se ilustra en la figura 2.4b.

El segundo tipo de tratamiento de imágenes, la binarización, conlleva utilizar un umbral sobre el cual, si el valor de un determinado píxel es mayor a él, el píxel se convierte en negro, y si es menor se convierte en blanco o viceversa. Normalmente, la binarización y otros tipos de métodos para el tratamiento de imágenes se aplican en imágenes que ya se han convertido a escala de grises. En la figura 2.4c se puede observar el proceso de binarización de una imagen.

Por último, la reducción de ruido o suavizado, se refiere a la aplicación de diversos filtros a una imagen con el objetivo de mejorar la visualización de algún objeto en particular, o simplemente para mejorar la calidad de la imagen. El filtro *gaussiano* [5] es uno de los filtros utilizados para realizar este tipo de tratamiento en una imagen. En dos dimensiones, siendo el caso de las imágenes, el filtro gaussiano ésta dado por la ecuación (2.1),

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (2.1)$$

donde x es la distancia desde el origen en el eje horizontal, y la distancia desde el origen en el eje vertical y σ es la desviación estándar de la distribución gaussiana.

³Conversión de una imagen a colores binarios (blanco y negro).

⁴Del inglés *Red, Green, Blue*; es la composición del color en términos de la intensidad de los colores primarios de la luz.

En la figura 2.4d se observa la aplicación de un filtro gaussiano a una imagen en escala de grises.

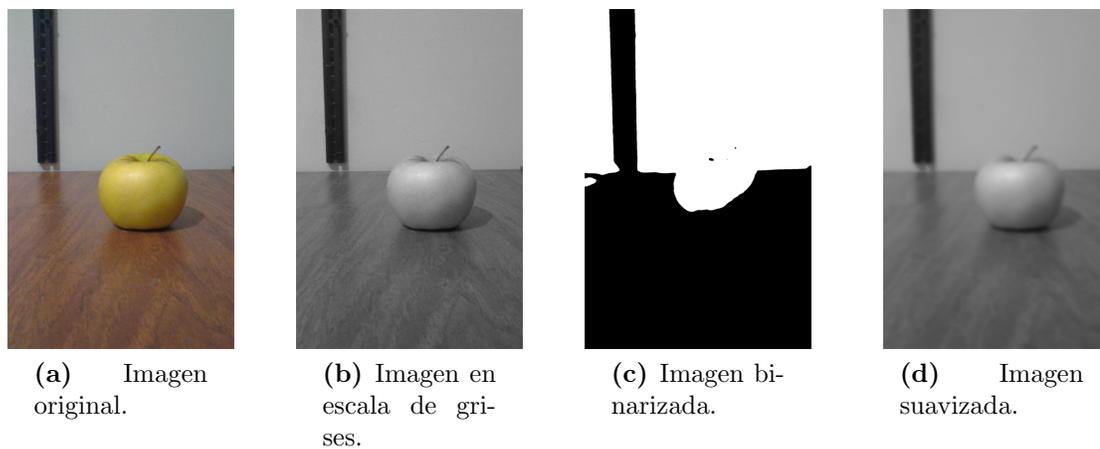


Figura 2.4 – Procesamiento de una imagen.

RECONOCIMIENTO DE OBJETOS

Dada una imagen que contiene uno o más objetos de interés (más su fondo) y un conjunto de etiquetas correspondientes a un conjunto de modelos conocidos, el sistema debe asignar las etiquetas correctamente a las regiones de la imagen donde se encuentran los objetos.

Una forma cualitativa de considerar la complejidad del problema de reconocimiento de objetos es considerando los siguientes factores:

- *Consistencia de la escena*: La complejidad de la escena (área donde se buscará el objeto); depende fuertemente de que las imágenes adquiridas durante el reconocimiento se encuentren en condiciones similares a las usadas como referencia. Bajo diversas condiciones, el desempeño de diferentes detectores de características es significativamente diferente.
- *Número de objetos en la imagen*: Si solo hay un objeto en la imagen, lo más

probable es que sea completamente visible. Incrementando el número de objetos, la probabilidad de que se cubran unos a otros incrementa.

Jain et al. [13] mencionan diversos algoritmos para el reconocimiento de objetos dado un sistema donde se conocen los modelos a buscar previamente.

RASTREO DE OBJETOS

El rastreo de objetos es una tarea importante en la visión computacional. En su forma más simple, puede ser definido como el problema de estimar la trayectoria de un objeto en movimiento durante una secuencia de imágenes [29].

Existen tres etapas clave en el rastreo de objetos en un video: detección de objetos de interés en movimiento, rastreo de dichos objetos en cada imagen sucesiva, y (en casos complejos) análisis de los rastros del objeto para reconocer su comportamiento y predecir futuros movimientos.

El rastreo de objetos tiende a ser complejo debido a:

- Ruido en las imágenes (amplificado por el movimiento).
- Movimiento complejo de los objetos.
- Formas complejas de objetos.
- Cambios en la iluminación.
- Requerimientos de procesamiento en tiempo real.

2.4 CÓDIGOS QR

Los códigos *QR* (del inglés *Quick Response*) son una forma avanzada de códigos de barras que pueden almacenar más de 1800 caracteres de texto en información

[16]. Con la llegada de los teléfonos inteligentes y las aplicaciones para escanear códigos QR, estos han crecido en popularidad debido a la facilidad para transferir información a un celular.

Un código QR permite almacenar diferentes tipos de información, incluyendo:

- Mensajes de texto plano (figuras 2.5a y 2.5b).
- Direcciones de paginas de internet (figura 2.5c).
- Información de contacto.
- Números de teléfono.
- Información geográfica.
- Un evento de calendario.



(a) Código con “7” almacenado.



(b) Código con “hola” almacenado.



(c) Código con “<http://www.uanl.mx>” almacenado.

Figura 2.5 – Ejemplos de códigos QR.

En la presente tesis, los códigos QR se utilizan como marcadores para superponer sobre ellos un determinado modelo tridimensional. Esto agrega un beneficio notable al proyecto ya que permite mantener un estándar para los marcadores, manejando únicamente códigos QR para dicha función. Además la capacidad de almacenar

información en un código QR, permite la opción de utilizar diferentes modelos tridimensionales para cada código QR distinto, utilizando la información decodificada del código para determinar que modelo utilizar.

En la figura 2.6 se puede observar un ejemplo de la detección y decodificación de códigos QR.



Figura 2.6 – Código QR detectado y decodificado.

CAPÍTULO 3

LITERATURA EXISTENTE

En este capítulo se presentan trabajos que contienen elementos similares al contenido de este trabajo, describiendo brevemente el contenido de cada uno de ellos. Al final se realiza una tabla comparativa identificando las características en común que comparten entre ellos.

3.1 TRABAJOS SIMILARES

En esta sección se resumen los trabajos relacionados más relevantes que se utilizan como referencia para la presente tesis.

INTERFACES DE REALIDAD AUMENTADA

Rohs [20] realiza una investigación sobre interfaces móviles de realidad aumentada que constan de componentes virtuales y componentes reales. En ella menciona técnicas de interacción espacial que capturan ópticamente el movimiento del dispositivo y su orientación con respecto a un marcador. Es relevante para la presente debido a que se maneja no solo el tema de realidad aumentada, sino además la detección de marcadores, factor importante en este trabajo.

Weilguny [26] aborda los aspectos de diseño relevantes en la realidad aumentada

enfocada en juegos, las consideraciones a tomar para aplicaciones de entretenimiento y la complejidad del diseño de una experiencia satisfactoria en este tipo de sistemas. El diseño de interfaces de realidad aumentada debe tomar consideraciones totalmente distintas a las de una interfaz normal en una computadora, pudiendo además ser combinada con diseño de interfaces móviles según sea el caso. Conocer los aspectos relevantes en el diseño de este tipo de aplicaciones ayuda a desarrollar mejores sistemas fáciles de usar tanto para desarrolladores como para usuarios.

JUEGOS DE REALIDAD AUMENTADA

Tan y Soh [24] presentan una reseña de diversos videojuegos de realidad aumentada, realizados por ambos la industria y la comunidad investigadora en el área. Conociendo diversas aplicaciones y juegos que existen tanto en la industria como investigación ayuda a conocer mejor el área de oportunidades que existe para la realidad aumentada.

Huynh et al. [11] muestran el juego *Art of Defense*, un juego cooperativo móvil que utiliza realidad aumentada. En el trabajo discuten los detalles técnicos del juego, la racionalidad detrás de su diseño así como el proceso que se siguió para obtener una buena experiencia de usuario. El proceso de diseño de un videojuego estratégico con realidad aumentada como lo es *Art of Defense* es totalmente distinto al proceso de un videojuego móvil normal, teniendo que tomar factores en cuenta como lo son: el dispositivo móvil usado, la calidad de la cámara del mismo, la interacción del usuario con la aplicación y con el mundo para realizar algún evento en el videojuego, etcétera.

Geiger et al. [8] describen un sistema que permite la integración de características de realidad aumentada con técnicas avanzadas de interacción para el desarrollo de juegos de realidad aumentada.

Broll et al. [2] realizan una reseña de los retos tecnológicos que se deben cumplir para realizar juegos de realidad aumentada, presentando una librería que los supera, probando además su acercamiento con la implementación de tres juegos de realidad aumentada.

Wetzel et al. [27] presentan una guía de diseño y desarrollo de juegos de realidad aumentada, especializada en juegos de estrategia.

APLICACIONES DE REALIDAD AUMENTADA

Wagner y Schmalstieg [25] describen en su trabajo una librería que permite la creación de aplicaciones de realidad aumentada en dispositivos móviles accesibles a usuarios.

Skwarek [21] presenta *CreatAR*, una aplicación de realidad aumentada para dispositivos móviles que permite a los usuarios crear contenido utilizando realidad aumentada basada en *geolocalización*¹.

Mata y Claramunt [14] introducen un sistema de realidad aumentada que proporciona facilidades de navegación, generación de itinerarios y servicios de entrega. El acercamiento del trabajo ofrece la posibilidad de combinar escenarios reales con representaciones digitales de lugares de interés y servicios para un itinerario dado.

3.2 CLASIFICACIÓN DE TRABAJOS SIMILARES

El cuadro 3.1 muestra en que categoría se puede comparar cada trabajo previamente mencionado con respecto a este trabajo. La totalidad de los trabajos contiene algún elemento que utilice programación con gráficos tridimensionales para crear un sistema de realidad aumentada. Esto demuestra que la habilidad para visualizar

¹Localización satélital utilizando coordenadas de latitud, longitud y en algunos casos altitud.

CAPÍTULO 4

SOLUCIÓN PROPUESTA

En este capítulo se presenta la descripción de la propuesta que se consideró para el problema de la implementación de un sistema de realidad aumentada, explorando la metodología usada para solucionar el problema, las herramientas que se usaron (tanto dispositivos, como programas y librerías de código abierto) y la teoría detrás de la implementación.

4.1 METODOLOGÍA

La metodología utilizada en el desarrollo del sistema de realidad aumentada es el llamado *modelado en cascada* definido como el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del sistema, de tal forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior [22].

El modelado en cascada definido para este trabajo consiste en un proceso secuencial separado en las fases de: análisis, diseño, implementación, experimentación (figura 4.1); por ser el caso de un proyecto de investigación, se omiten fases como mantenimiento y pruebas, reemplazándolas por la etapa de experimentación.

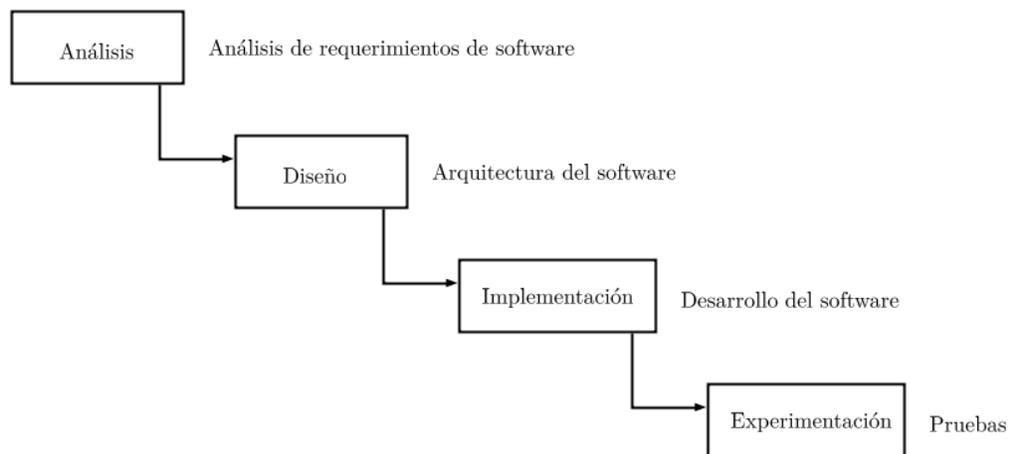


Figura 4.1 – Diagrama de la metodología utilizada.

4.1.1 ANÁLISIS

En la etapa de análisis se hace un estudio del proyecto, definiendo los requerimientos del sistema, el alcance que éste tendrá, las herramientas a usar, etcétera.

ALCANCE

El alcance del proyecto se refiere al punto al que se desea llegar en el desarrollo del sistema. En este caso, el alcance corresponde al objetivo general del trabajo, definido como:

“Implementación de un sistema que mediante la detección y decodificación de códigos QR, realice la superposición de modelos 3D sobre ellos”.

REQUERIMIENTOS

Los requerimientos definidos para el sistema propuesto pueden definirse en base del alcance del proyecto. Para desarrollar el sistema propuesto se puede dividir el problema en tres etapas:

1. Procesamiento de imágenes.
2. Detección y decodificación de códigos QR.
3. Superposición de modelos 3D.

La primera etapa consistiría en la extracción y procesamiento de imágenes a partir de un video o de una cámara. Por ejemplo en el caso de un video, las imágenes se extraen secuencialmente hasta que no existan más. A cada una de ellas se le aplican múltiples filtros para disminuir la cantidad de información en ellas y mejorar la visualización de los posibles códigos QR que existan, de tal forma que la detección y decodificación de los mismos sea más eficiente.

En la etapa de detección y decodificación de códigos se llevaría a cabo todo el proceso desde la localización de un posible código QR, hasta la decodificación del mismo para obtener la información almacenada en él.

Finalmente en la etapa de superposición de modelos 3D se realizaría la superposición de modelos sobre los códigos previamente detectados, únicamente en aquellos cuya información almacenada los relacione con un modelo existente en la base de datos del sistema.

Entonces en base a lo anterior, los requerimientos para el sistema son:

- Extracción de imágenes desde un video o cámara.
- Procesamiento de imágenes extraídas.
 - Conversión a escala de grises.
 - Escalamiento de la imagen (si es muy grande).
 - Binarización de la imagen.
- Detección de códigos QR.

- Decodificación de códigos QR detectados.
- Búsqueda de modelos en base a información de códigos QR.
- Superposición de modelos 3D.

Con la definición de los requerimientos es posible comenzar a pensar en que tipo de herramientas se pueden utilizar, tanto el hardware necesario para desarrollar el sistema como las posibles librerías de sistema que puedan ayudar a realizar tareas secundarias que forman parte de la implementación del sistema.

HERRAMIENTAS

En el cuadro 4.1 se muestran las herramientas de hardware y *software*¹ utilizadas en el presente trabajo.

Cuadro 4.1 – Herramientas de hardware y software utilizadas.

Hardware	
Hardware	Uso
Computadora (3.1 Ghz, 6 núcleos y 8 GB de memoria RAM)	Plataforma principal para el desarrollo del sistema de realidad aumentada.
Samsung Galaxy Tab 2	Dispositivo utilizado para la grabación de video y captura de imágenes para pruebas.
Software	
Software	Uso
Python 2.7	Lenguaje de programación utilizado para el desarrollo del sistema. https://www.python.org/download/releases/2.7/
OpenCV 2.4.8	Librería para implementación de técnicas de visión computacional. http://opencv.org/downloads.html
ZBar	Librería para la detección y decodificación de códigos QR. http://zbar.sourceforge.net/download.html
Pygame 1.9.1	Interfaz entre OpenCV y el visualizador 3D. http://www.pygame.org/download.shtml

¹Programas de computadora que hacen posible la realización de tareas específicas.

4.1.2 DISEÑO

La etapa de diseño consiste en la definición de las especificaciones, la arquitectura, módulos y otras consideraciones necesarias para planear correctamente el sistema que se desarrolla.

ARQUITECTURA

La arquitectura del sistema es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad.

Para implementar el sistema propuesto cumpliendo tanto los requisitos funcionales como los no funcionales mencionados anteriormente, se define una secuencia de pasos a seguir de tal forma que el problema se convierta en simples problemas de decisión. Entonces es posible definir el flujo del programa, empezando con la lectura de una entrada de video (archivo multimedia o cámara web) desde la cual se extrae una imagen utilizando la librería `OpenCV`. Para el caso de un archivo de video se extraen las imágenes secuencialmente en orden hasta terminar el video, en cambio, en el caso de una cámara web se extraen imágenes hasta que el sistema termine su ejecución o la cámara web sea detenida.

En cualquiera de los dos casos, la imagen extraída se procesa utilizando `OpenCV` para consecuentemente buscar los códigos QR disponibles en ella. Si alguno de los códigos hace referencia a alguno de los modelos almacenados, estos se cargan en la ventana de `Pygame` (mostrando la imagen original, no la procesada) efectivamente aumentando la realidad. El proceso se puede ver en la figura 4.2.

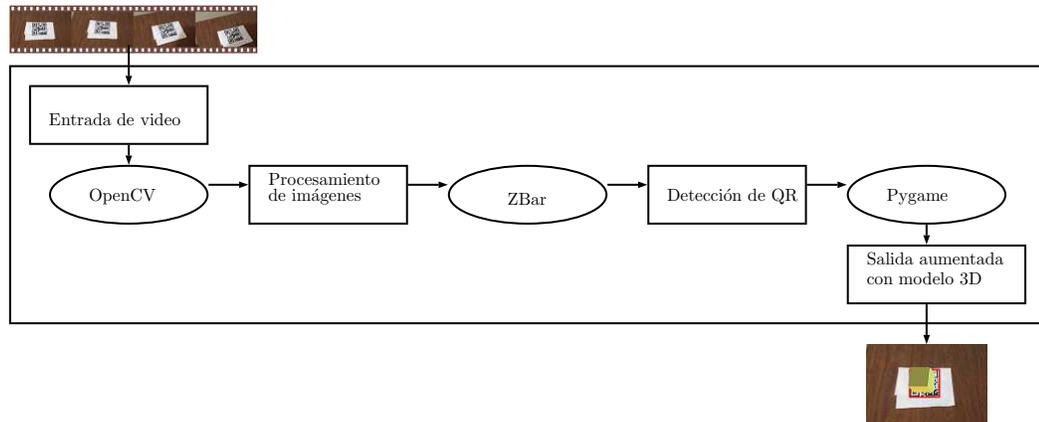


Figura 4.2 – Arquitectura del sistema propuesto.

DIAGRAMAS DE CLASES

El diagrama de clases es utilizado para determinar la interacción existente entre las distintas clases que existen dentro de un sistema. Por simplicidad de diseño y para facilitar el entendimiento del funcionamiento del sistema, el uso de todas las librerías externas se encuentra envuelto sobre una capa de abstracción² para limitar el código a lo necesario, ignorando otras funciones de las librerías que no son requeridas. Esto se puede observar en la figura 4.3, donde la clase `QR` utiliza la librería `Zbar` únicamente para realizar dos funciones: escaneo de un solo código QR (devolviendo el primero encontrado si hay más de uno) y escaneo de todos los códigos QR detectados.

La clase `CVision` utiliza la librería `OpenCV` para realizar las funciones de procesamiento de imágenes, y en este caso solo permite realizar aquellas que son útiles para el proyecto, siendo éstas:

- Escalamiento de una imagen.
- Conversión a escala de grises.
- Binarización.

²Exclusión de todos los elementos que no tienen cabida en el sistema.

Por último la clase *3D* se encarga de realizar la superposición de los modelos 3D. Ésta se limita a crear una ventana donde se observarán las imágenes extraídas desde la entrada de video, y a superponer modelos en las posiciones adecuadas dependiendo de la decodificación de códigos QR.

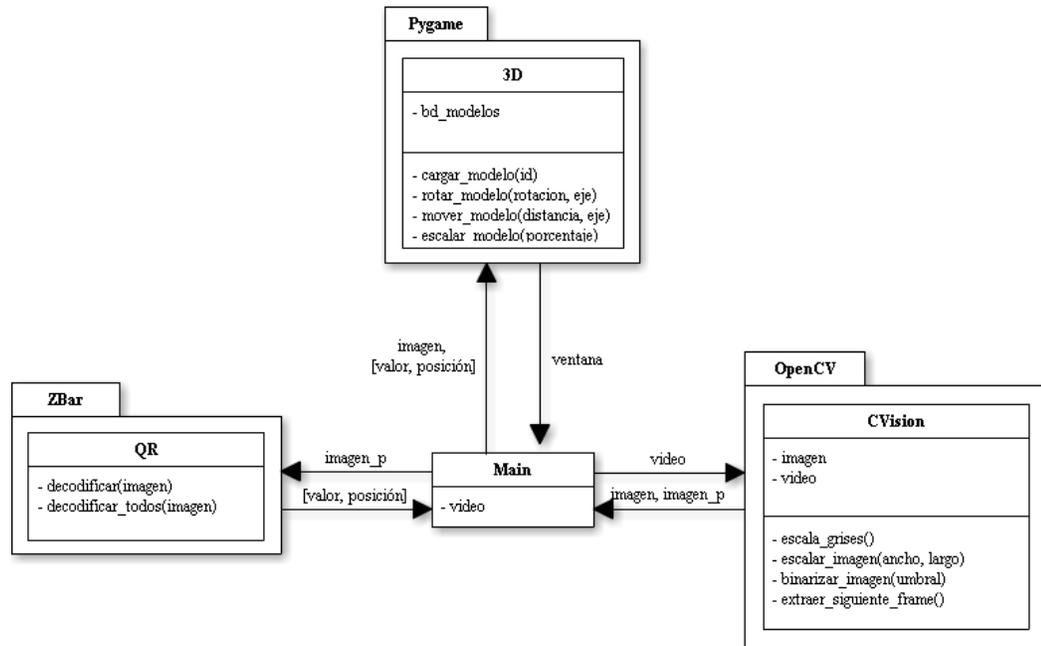


Figura 4.3 – Diagrama de clases del sistema.

4.2 MARCO TEÓRICO

En esta sección se describen los elementos que integran la implementación del sistema, ordenados secuencialmente en el orden en el que éstos se utilizan en el sistema.

4.2.1 VISIÓN COMPUTACIONAL

El sistema toma como entrada una secuencia de imágenes (video, cámara), las cuales son pre-procesadas utilizando técnicas de visión computacional para facilitar la detección y decodificación de los códigos QR usados como marcadores.

El procedimiento usado es el siguiente:

1. Lectura de una imagen desde la secuencia de imágenes.
2. Ajuste de tamaño de la imagen a un óptimo.
3. Conversión a escala de grises.
4. Binarización.

LECTURA DE UNA IMAGEN

El proceso consiste en extraer una imagen desde una secuencia de imágenes. En el caso de un video, estos consisten en un conjunto de imágenes por lo cual la tarea es simplemente obtener cada imagen individual secuencialmente hasta terminar el video.

AJUSTE DEL TAMAÑO DE UNA IMAGEN

Si las imágenes son muy grandes, para mejorar la eficiencia de la detección y decodificación de códigos QR se busca escalarlas a un tamaño adecuado para visualizar correctamente la imagen, sin tener una resolución innecesariamente grande. Para realizar el escalamiento de la imagen se utiliza un escalamiento bilineal que utiliza interpolación lineal para estimar el valor de un pixel intermedio a partir de un pixel inicial y uno final.

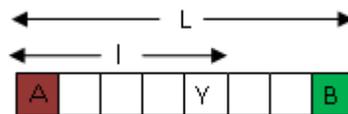


Figura 4.4 – Interpolación lineal.

Por ejemplo, en la figura 4.4, Y puede ser calculada utilizando los valores conocidos

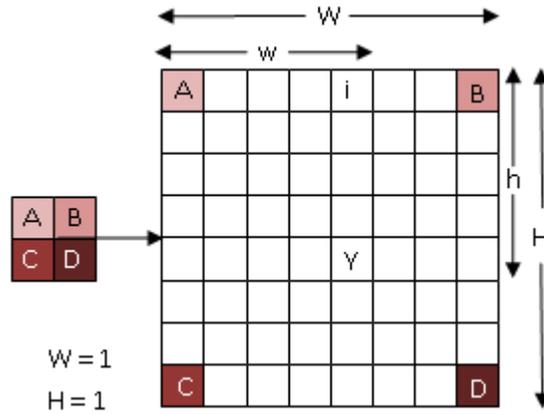


Figura 4.5 – Interpolación bilineal.

de A, B, L y l , realizando una interpolación lineal:

$$\frac{Y - A}{l} = \frac{B - A}{L} \rightarrow Y = \frac{l(B - A)}{L}. \quad (4.1)$$

En el caso de una imagen, sin embargo, se tienen dos dimensiones por lo que la interpolación se hace en dos dimensiones (alto y ancho) como se observa en la figura 4.5.

Utilizando interpolación lineal, primero se obtienen las ecuaciones de i, j, Y :

$$\frac{i - A}{W} = \frac{B - A}{W} \rightarrow i = A + \frac{w(B - A)}{W} \rightarrow i = A + w(B - A), \quad (4.2)$$

$$\frac{j - C}{W} = \frac{D - C}{W} \rightarrow j = C + \frac{w(D - C)}{W} \rightarrow j = C + w(D - C), \quad (4.3)$$

$$\frac{Y - i}{h} = \frac{j - i}{H} \rightarrow Y = i + \frac{h(j - i)}{H} \rightarrow Y = i + h(j - i). \quad (4.4)$$

Sustituyendo i, j en la última ecuación,

$$\begin{aligned} Y &= A + w(B - A) + h(C + w(D - C) - (A + w(B - A))), \\ Y &= A(1 - w)(1 - h) + B(w)(1 - h) + C(h)(1 - w) + D(wh). \end{aligned} \quad (4.5)$$

Con esta última ecuación se calcula el valor de cualquier pixel restante en la imagen, reemplazando w, h por los valores correspondientes al pixel que se busca calcular.

CONVERSIÓN A ESCALA DE GRISES

La conversión a escala de grises corresponde a la transición de una imagen en formato *RGB* a una imagen que contiene un solo canal que denota la intensidad del color de un pixel, normalmente desde 0 hasta 255. Este proceso aumenta la eficiencia de la detección y decodificación de los códigos QR debido a que disminuye la cantidad de información disponible en la imagen, al disminuir la cantidad de canales por pixel, por lo cual el procesamiento del algoritmo de detección es más rápido.

Para calcular el valor de un pixel en escala de grises dados los valores de sus canales *RGB*, se utiliza la ecuación:

$$Y = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B, \quad (4.6)$$

donde Y es el nuevo valor del pixel en escala de grises, R es la intensidad del color rojo en el pixel original, G la intensidad del color verde y B la intensidad del color azul.

BINARIZACIÓN

El proceso de binarización como se explicó anteriormente consiste en convertir los pixeles de una imagen a valores binarios, normalmente 0 o 255 (negro o blanco). Esto se realiza mediante la comparación del valor de un pixel en escala de grises con un umbral de decisión.

Por simplicidad algunas veces se compara contra 127, o el equivalente valor intermedio que un pixel puede tener en una escala de grises, pero para mejorar el desempeño

de la binarización en ambientes donde la iluminación no es constante, se utilizan algoritmos más complejos como es el *método de Otsu* [17]. El método de Otsu consiste en examinar el histograma los valores de los píxeles en una imagen en escala de grises para encontrar dos picos y en base a la mitad de ellos, asignar el umbral. La figura 4.6 muestra el funcionamiento del método Otsu.

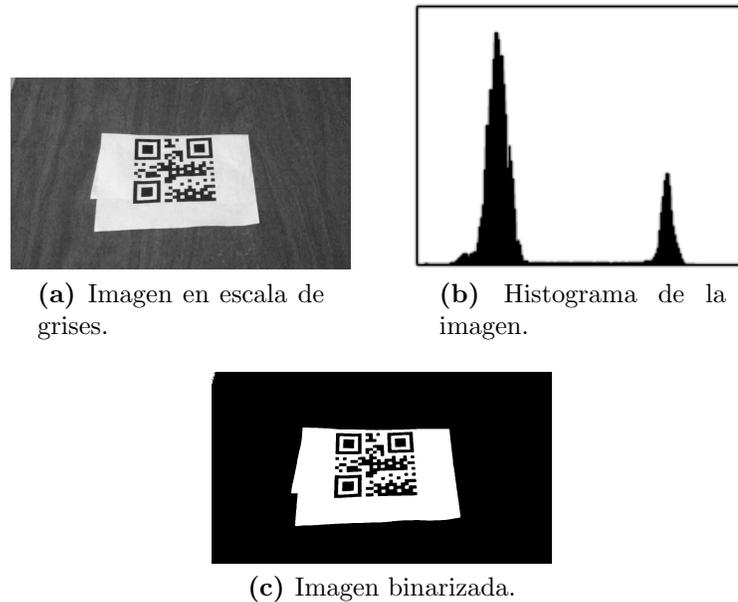


Figura 4.6 – Método Otsu para la binarización de una imagen.

4.2.2 PROGRAMACIÓN TRIDIMENSIONAL

Para cargar los modelos 3D se utilizan archivos Wavefront OBJ (figura 4.7), que consisten en la descripción de:

- Coordenadas de los vértices.
- Mapeo de las texturas.
- Definición de las normales.
- Definición de las caras.

COORDENADAS DE LOS VÉRTICES

Especifican un vértice dadas tres coordenadas x, y, z . Este vértice es llamado implícitamente dependiendo del orden en el que fue declarado en el archivo. Así el primer vértice declarado en un archivo es “1”, el segundo “2”, etcétera.

MAPEO DE LAS TEXTURAS

Definen el mapeado de la textura de un vértice. Corresponden a valores flotantes entre cero y uno que determinan como se hará el mapeado de la textura. No son obligatorios.

DEFINICIÓN DE LAS NORMALES

Describen un vector normal, usado para determinar a qué dirección apuntan las caras. En ocasiones no son usados ya que estos se calculan utilizando el orden de los vértices.

DEFINICIÓN DE LAS CARAS

Define un polígono formado por los vértices especificados. Es posible utilizar tantos vértices como sean necesarios. Para referenciar un vértice se utiliza su índice desde el archivo (asignado en orden de definición). Se asume que los vértices de la cara se especifican en orden inverso a las manecillas del reloj, por lo cual si no se definen las normales, para invertir la dirección de las caras es posible definir los vértices en orden de las manecillas del reloj.

g cube	# Normales	# Caras
# Vertices	vn 0.0 0.0 100.0	f 1//2 7//2 5//2
v 0.0 0.0 0.0	vn 0.0 0.0 -100.0	f 1//2 3//2 7//2
v 0.0 0.0 100.0	vn 0.0 100.0 0.0	f 1//6 4//6 3//6
v 0.0 100.0 0.0	vn 0.0 -100.0 0.0	f 1//6 2//6 4//6
v 0.0 100.0 100.0	vn 100.0 0.0 0.0	f 3//3 8//3 7//3
v 100.0 0.0 0.0	vn -100.0 0.0 0.0	f 3//3 4//3 8//3
v 100.0 0.0 100.0		f 5//5 7//5 8//5
v 100.0 100.0 0.0		f 5//5 8//5 6//5
v 100.0 100.0 100.0		f 1//4 5//4 6//4
		f 1//4 6//4 2//4
		f 2//1 6//1 8//1
		f 2//1 8//1 4//1

Figura 4.7 – Cubo descrito en el formato Wavefront OBJ.

El archivo de la figura 4.7 cargado mediante un visualizador 3D describe la geometría de un cubo, como se ve en la figura 4.8.

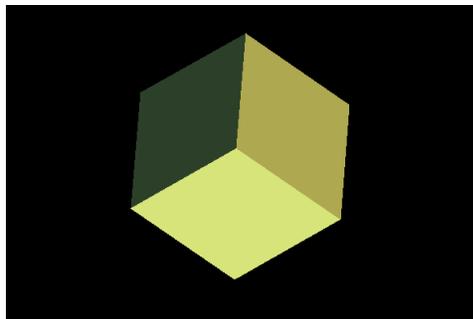


Figura 4.8 – Cubo creado a partir de un archivo Wavefront OBJ.

4.3 IMPLEMENTACIÓN

En esta sección se describirá paso a paso el método de implementación propuesto para el desarrollo del sistema de realidad aumentada.

4.3.1 ALGORITMO

El algoritmo y su correspondiente diagrama de flujo (figura 4.9) propuestos para la implementación del sistema consisten en:

1. Leer una imagen de un video o cámara. Si no hay imágenes terminar.
2. Procesar la imagen con técnicas de visión computacional para facilitar la detección de códigos QR.
3. Buscar posibles códigos QR.
 - 3.1. Decodificar códigos QR detectados.
 - 3.1.1. Si algún código hace referencia a un modelo 3D, cargar el modelo sobre la posición del código QR.
4. Volver al paso 1.

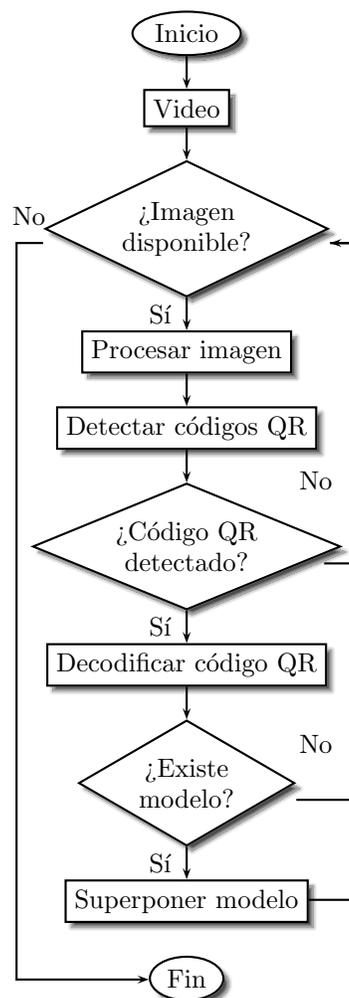


Figura 4.9 – Diagrama de flujo del sistema propuesto.

Como se puede ver en la figura 4.9, después del procesamiento de la imagen actual, el problema queda reducido a una serie de decisiones. Si se detecta algún código QR, se procede decodificar, de otra forma se extrae la siguiente imagen. Si se decodifica exitosamente algún código QR detectado se procede a buscar un modelo correspondiente, de otra forma se extrae la siguiente imagen. Finalmente si existe un modelo para el valor de algún código QR detectado, éste se superpone en la posición del código detectado. En cualquier momento al extraer la siguiente imagen, si no se encuentra alguna, se detiene la ejecución.

4.3.2 PROCESAMIENTO DE IMÁGENES

Como se ha mencionado anteriormente, la etapa de procesamiento de imágenes es la que conlleva la aplicación de filtros a cada imagen individual de una secuencia de imágenes con el objetivo de disminuir la cantidad de información en la imagen y con ello aumentar la eficiencia de la detección y decodificación de códigos QR.

LECTURA DE IMAGEN

A partir de un video o cámara de entrada se extraen imágenes individuales para realizar el proceso de detección de códigos QR y superposición de modelos 3D. De esta manera si se tiene un video como en la figura 4.10a, la primera imagen extraída es la figura 4.10b.



(a) Video de ejemplo



(b) Primer imagen extraída del video

Figura 4.10 – Extracción de una imagen de un video.

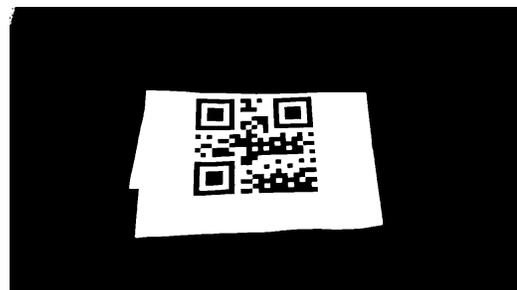
CONVERSIÓN A ESCALA DE GRISES Y BINARIZADO

Al contar con una imagen extraída desde el video de entrada, lo siguiente es convertir la imagen a escala de grises con la ecuación 4.6. El resultado de la conversión a escala de grises usando la imagen extraída se puede observar en la figura 4.11a.

Otro filtro útil pero no absolutamente necesario es el binarizado de la imagen, aplicado como se menciona en la sección anterior. La aplicación de este filtro sobre la imagen en escala de grises se puede observar en la figura 4.11b.



(a) Escala de grises



(b) Binarizado

Figura 4.11 – Conversión a escala de grises y binarización de la imagen extraída.

DETECCIÓN Y DECODIFICACIÓN DE CÓDIGOS QR

Después de aplicar los filtros se procede a buscar posibles códigos QR en la imagen procesada. Para ello se utiliza la librería Zbar especializada en la decodificación de códigos de barras y QR, donde el proceso realizado por la misma es:

- Binarizado la imagen.
- Agrupado de pixeles en módulos.
- Detección de patrones de búsqueda y alineación de los códigos QR.
- Rotación de la imagen y corrección de la distorsión.
- Corrección de error utilizando el código Solomon-Reed [28].
- Extracción de los datos a partir de los patrones horizontales y verticales.

El resultado de decodificar el código QR de la primera imagen extraída se puede observar en la figura 4.12.



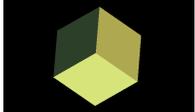
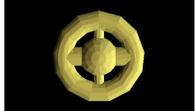
Figura 4.12 – Código QR detectado y decodificado.

SUPERPOSICIÓN DE MODELOS

Si se detectó algún código QR y su decodificación fue exitosa, el valor obtenido de la decodificación se utiliza como referencia para cargar el modelo 3D correspondiente a ese código QR sobre la posición del código.

De esta manera, suponiendo que se tiene una tabla de referencia como el cuadro 4.2, el modelo a cargar para el código QR detectado en la figura 4.12 es “1.obj” (figura 4.13).

Cuadro 4.2 – Selección de modelos tridimensionales.

Código QR	Valor almacenado	Archivo correspondiente	Modelo 3D
	“1”	“1.obj”	
	“2”	“2.obj”	
	“3”	“3.obj”	

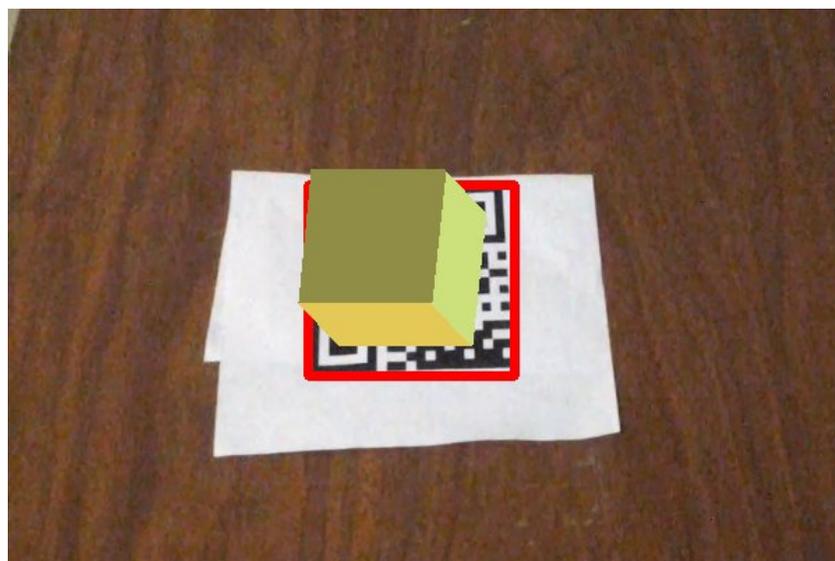


Figura 4.13 – Modelo superpuesto en el código detectado.

CAPÍTULO 5

EXPERIMENTOS

En este capítulo se definirán los diferentes experimentos que se pondrán a prueba para conocer el comportamiento del sistema en dos circunstancias:

- En un estado por defecto, es decir, utilizando un solo código QR y evaluando su desempeño al incrementar la complejidad de los modelos.
- Cuando se utilizan múltiples códigos QR.

Para cada caso, se realizarán mediciones de tiempo en segundos para cuatro procesos:

1. Procesamiento de la imagen.
2. Decodificación de códigos QR.
3. Superposición de modelo 3D.
4. Tiempo total.

Se presenta una gráfica con los resultados obtenidos para cada uno de los casos.

5.1 UN SOLO CÓDIGO QR

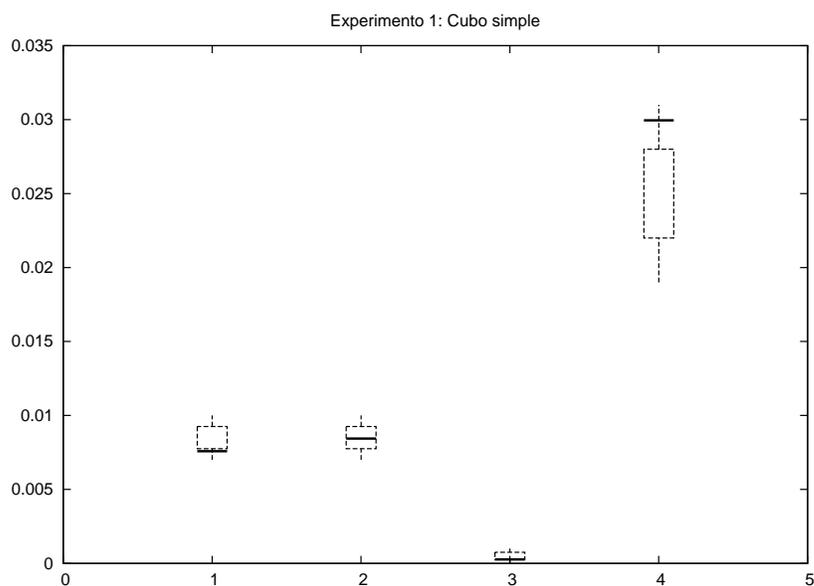
El ejemplo de uso por defecto para el sistema es localizando un solo código QR, sirviendo este caso de referencia para comparar con los demás casos de estudio donde se evalúa el sistema en ambientes más exigentes.

MODELO TRIDIMENSIONAL DE UN CUBO

En un inicio, se utiliza el conocido modelo 3D de un cubo para estudiar el desempeño del sistema en uno de los casos más simples, esto con el objetivo de tener una referencia útil para poder comparar más adelante con casos más complejos. Los resultados pueden verse en la figura 5.1.



(a) Captura del sistema



(b) Resultados

Figura 5.1 – Modelo 3D de un cubo con un solo código QR.

ANÁLISIS

Para este caso los resultados son adecuados, el procesamiento general tarda alrededor de 0.03 segundos en promedio, por lo cual es aceptable. Al ser un solo código QR y por lo tanto un modelo 3D, el tiempo que se tarda en dibujar el modelo es bastante bajo, siendo aún menor al tiempo que tarda el procesamiento de la imagen y la detección del código.

MODELO TRIDIMENSIONAL DE UNA RUEDA

Ahora, se utiliza un modelo un poco más complejo, que cuenta con una mayor cantidad de vértices y caras. Este se puede observar en la figura 5.2.



(a) Captura del sistema

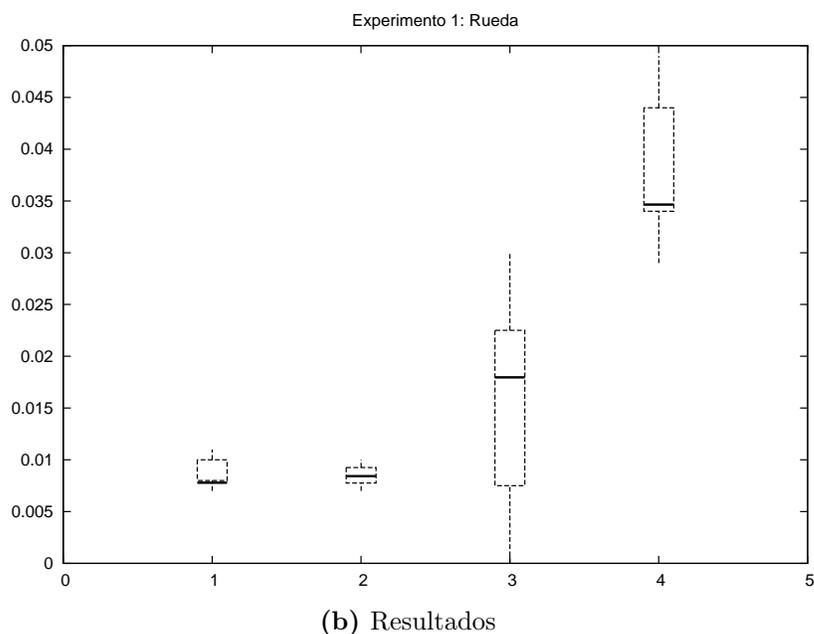


Figura 5.2 – Modelo 3D de una rueda con un solo código QR.

ANÁLISIS

Etapas como el procesamiento de la imagen y detección del código QR no varían debido a que se utiliza el mismo video de prueba en el primer experimento. Se puede observar un ligero incremento en la superposición del modelo, pero nada mayor aún,

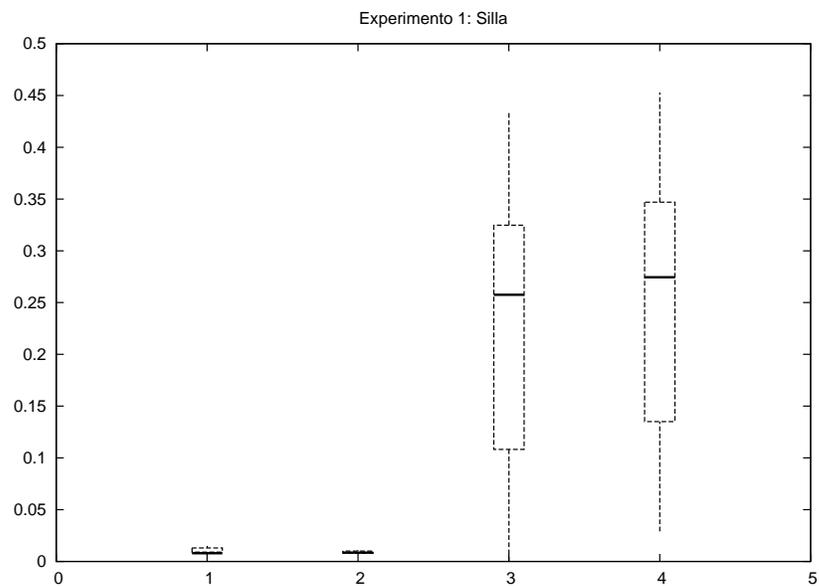
siendo aceptable el tiempo global.

MODELO TRIDIMENSIONAL DE UNA SILLA

Por último para el primer experimento, se utiliza un modelo mucho más complejo que cuenta con una gran cantidad de vértices. Éste es el modelo de una silla, que puede observarse en la figura 5.3.



(a) Captura del sistema



(b) Resultados

Figura 5.3 – Modelo 3D de una silla con un solo código QR.

ANÁLISIS

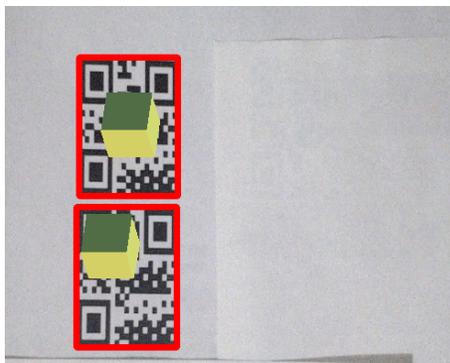
De nuevo, ambos el procesamiento de la imagen y la detección del código QR se ignoran porque son constantes en este experimento. Pero ahora se puede observar un incremento bastante alto en el tiempo que se tarda en realizar la sobreposición del modelo 3D, incrementando de 0.0175 segundos a 0.27 segundos, que es más de diez veces. Esto indica que existe un área de oportunidad para optimizar la visualización de los modelos 3D, ya que el modelo utilizado no es uno muy complejo y este tiempo no es aceptable.

5.2 MÚLTIPLES CÓDIGOS QR

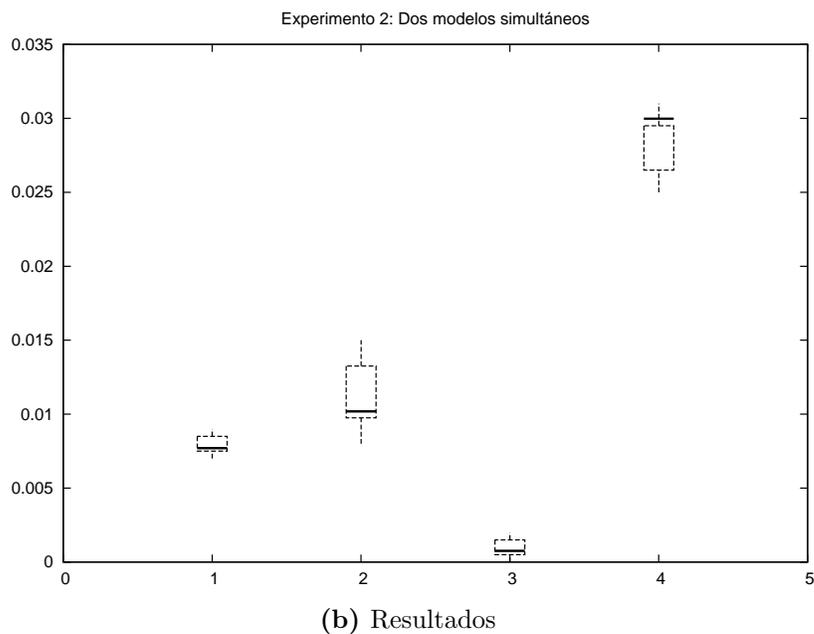
Otro caso interesante a estudiar es el desempeño del sistema utilizando múltiples códigos QR. Ya se comprobó por el experimento anterior que el sistema no está apto para trabajar con modelos complejos, por lo cual para este experimento solo se utilizará el modelo más simple (el cubo), utilizándolo múltiples veces.

DOS CÓDIGOS QR

Para empezar, se utiliza un video en donde se graban dos códigos QR por algunos momentos. Ambos códigos hacen referencia al mismo modelo del cubo por las razones especificadas anteriormente. Los resultados del experimento se visualizan en la figura 5.4.



(a) Captura del sistema

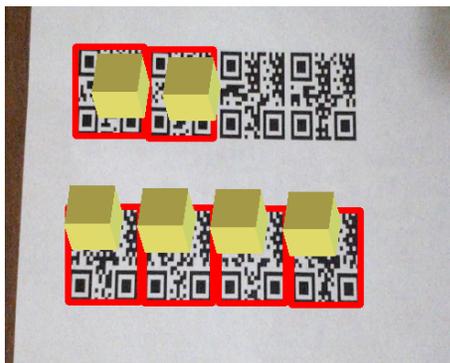
**Figura 5.4** – Superposición de dos modelos simultáneos.

ANÁLISIS

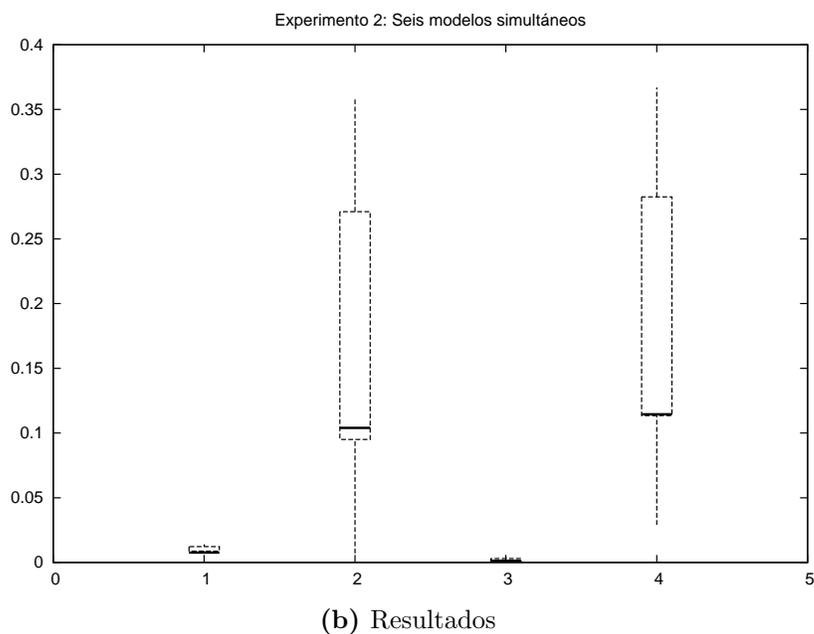
A diferencia del experimento anterior, en éste varía la entrada de video para cada caso. Con dos códigos a comparación con el experimento uno, el tiempo de decodificación de códigos QR incrementa al haber aumentado el número de códigos presentes, y debido a que se utilizan los cubos como modelos 3D, el tiempo que se tarda en realizar la superposición de los modelos es bajo aunque se esté trabajando con dos códigos.

SEIS CÓDIGOS QR

Observando los resultados anteriores es de esperar que al incrementar los códigos QR el proceso que incrementará su tiempo de ejecución es el de decodificación de códigos QR. Los resultados del experimento se visualizan en la figura 5.5.



(a) Captura del sistema



(b) Resultados

Figura 5.5 – Superposición de seis modelos simultáneos.

ANÁLISIS

En este experimento está claro el incremento del tiempo de procesamiento de la decodificación de los códigos QR. Éste llega a formar casi la totalidad del tiem-

po de procesamiento para cada imagen, lo cual es bastante deficiente, por lo que aquí también surge una oportunidad de optimización para mejorar el proceso de decodificación de los códigos QR.

CAPÍTULO 6

CONCLUSIONES

En este capítulo se concluye el presente trabajo, realizando comentarios finales al respecto del proyecto implementado, además de algunas menciones a las contribuciones que el proyecto pretende realizar y el posible trabajo a futuro.

6.1 CONTRIBUCIONES

La tesis “*Uso de marcadores para la superposición de modelos tridimensionales en imágenes*” presenta una forma de implementación de un sistema de realidad aumentada utilizando códigos QR como marcadores. El método de implementación combina técnicas de visión computacional como lo son la conversión a escala de grises y binarización de imágenes para lograr la detección y decodificación de los mismos y programación tridimensional para lograr la superposición de los modelos en las imágenes.

El proceso consiste en tomar una secuencia de imágenes, donde para cada imagen se buscan los códigos QR existentes en ella. Cuando se detecta un código QR, al decodificarlo el valor obtenido se utiliza como referencia para obtener el modelo 3D adecuado. Este proceso se repite para cada código QR detectado en cada imagen de la secuencia.

Se reportan experimentos realizados a la implementación para estudiar el comportamiento del sistema cuando se utilizan modelos complejos, así como cuando se utilizan múltiples códigos QR simultáneamente. De estos experimentos se puede concluir que existen dos áreas de oportunidad para mejorar la implementación, debido a que el uso de `Pygame` para la visualización de los modelos 3D es deficiente, así como la decodificación de códigos QR mediante la librería `Zbar`.

6.2 TRABAJO A FUTURO

Un aspecto que necesita ser dirigido es la visualización de modelos 3D. El enfoque del proyecto se realizó fuertemente hacia el estudio de la implementación de sistema de realidad aumentada habilitado mediante el uso de códigos QR, por lo que el visualizador de modelos 3D no está implementado en algún motor especializado de modelos 3D, sino que se utiliza un simple visualizador 2D con funciones de dibujo básicas para crear polígonos, y a partir de ellos crear modelos 3D.

Otra mejora a la implementación a considerar es el rastreo de códigos QR detectados. El sistema busca por códigos QR en cada imagen de una secuencia de imágenes, por lo cual si de una imagen a otra (consecutivas una a otra en la secuencia) no se localiza el código por cualquier razón, se visualiza un parpadeo en el modelo 3D debido a que no se encontró el código QR en esa imagen en específico, aún cuando una imagen antes se había encontrado. Esto es posible de mejorar realizando un rastreo de los códigos QR detectados (solamente los que hagan referencia a modelos 3D, por supuesto) desde que son localizados, buscando posibles incidencias en imágenes futuras dentro de la secuencia. Sin embargo, un experimento a considerar para esta implementación es la complejidad en cuanto a tiempo, debido a que para cada código QR detectado se tendría que realizar un rastreo además de su decodificación para cada imagen en una secuencia de imágenes.

Debido a que el enfoque del proyecto era dedicado al estudio de un tipo de implementación de sistema de realidad aumentada, y no a las implicaciones en usabilidad de el mismo, no se tomó en cuenta la implementación de la solución propuesta para dispositivos móviles, lo cual podría muy bien considerarse para un trabajo a futuro, habilitando una gran cantidad de aplicaciones. Además, para poder estudiar las capacidades de la implementación de aplicaciones con el sistema propuesto en este trabajo, otro posible trabajo a futuro podría ser la implementación de una aplicación que haga uso de esta solución para interactuar de alguna manera con usuarios y realizar otras tareas simultáneas.

BIBLIOGRAFÍA

- [1] Xavier Boix Bosch, Horesh Ben Shitrit, Pascal Fua y Gemma Roig Noguera. Conditional random fields for multi-camera object detection. En *Computer Vision (ICCV), 2011 IEEE International Conference on*, págs 563–570. IEEE, 2011.
- [2] Wolfgang Broll, Jan Ohlenburg, Irma Lindt, Iris Herbst y Anne-Kathrin Braun. Meeting technology challenges of pervasive augmented reality games. En *Proceedings of 5th ACM SIGCOMM Workshop on Network and System Support for Games, NetGames '06*, pág 28. ACM, Nueva York, NY, EE.UU., 2006. ISBN 1-59593-589-4. doi:10.1145/1230040.1230097.
- [3] Jia Deng, Jonathan Krause y Li Fei-Fei. Fine-Grained Crowdsourcing for Fine-Grained Recognition. En *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, págs 580–587. IEEE, Portland, Oregon, EE.UU., 2013.
- [4] The Economist. Reality, improved. 2009. <http://www.economist.com/node/14299602>.
- [5] Robert Fisher, Simon Perkins, Ashley Walker y Erik Wolfart. Gaussian smoothing. 2004. <http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>.
- [6] Luke Fletcher, Nicholas Apostoloff, Jason Chen y Alexander Zelinsky. Computer

- vision for vehicle monitoring and control. págs 67–72. Australian Conference on Robotics and Automation, Sydney, Australia, 2001.
- [7] Germán Amaya Franky. Los entornos virtuales de simulación de la realidad, espacios vistos como ejes que permiten situar el aprendizaje dentro de un contexto institucionalizado de educación. 2005. http://campus.usal.es/~teoriaeducacion/rev_numero_07/n7_art_gaf.htm.
- [8] Christian Geiger, Joerg Stoecklein, Florian Klompmaker y Robin Fritze. Development of an Augmented Reality Game by Extending a 3D Authoring System. En *Proceedings of the International Conference on Advances in Computer Entertainment Technology, ACE '07*, págs 230–231. ACM, Nueva York, NY, EE.UU., 2007. ISBN 978-1-59593-640-0. doi:10.1145/1255047.1255099.
- [9] Pawel Gutowski. Computer vision — people counting algorithm. 2009. <https://www.youtube.com/watch?v=TPjmsha6gWY>.
- [10] Weidong Huang, Leila Alem y Mark A. Livingston. *Human Factors in Augmented Reality Systems*. Berlín / Heidelberg, Alemania, 2012.
- [11] Duy-Nguyen Ta Huynh, Karthik Raveendran, Yan Xu, Kimberly Spreen y Blair MacIntyre. Art of Defense: A Collaborative Handheld Augmented Reality Board Game. En *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games, Sandbox '09*, págs 135–142. ACM, Nueva York, NY, EE.UU., 2009. ISBN 978-1-60558-514-7. doi:10.1145/1581073.1581095.
- [12] Oculus VR Inc. Oculus Rift - Virtual Reality Headset for 3D Gaming — Oculus VR. 2014. <http://www.oculusvr.com/>.
- [13] Ramesh Jain, Rangachar Kasturi y Brian G. Schunck. *Machine Vision*. McGraw-Hill, Inc., Nueva York, NY, EE.UU., 1995. ISBN 0-07-032018-7.

- [14] Felix Mata y Christophe Claramunt. Augmented Navigation in Outdoor Environments. En *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL'13*, págs 514–517. ACM, Nueva York, NY, EE.UU., 2013. ISBN 978-1-4503-2521-9. doi:10.1145/2525314.2525319.
- [15] Carlos González Morcillo, David Vallejo Fernández, Javier A. Albusac Jiménez y José Jesús Castro Sánchez. *Realidad Aumentada. Un Enfoque Práctico con ARToolkit y Blender*. Bubok Publishing S.L., Ciudad Real, España, 2009. ISBN 978-84-686-1151-8.
- [16] Stanford School of Medicine. Barcodes and QR codes. 2009. <http://med.stanford.edu/irt/teaching/barcode.html>.
- [17] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. En *Transactions on Systems, Man, and Cybernetics Vol. SMC-9, No. 1*. IEEE, 1979.
- [18] Antonio Painn. La Realidad Aumentada y los nuevos contenidos. 2014. <http://www.antoniopainn.com/la-realidad-aumentada-y-los-nuevos-contenidos/>.
- [19] Gerhard Reitmayr y Dieter Schmalstieg. *Location Based Applications for Mobile Augmented Reality*. AUIC '03. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 2003. ISBN 0-909925-96-8.
- [20] Michael Rohs. Marker-based embodied interaction for handheld augmented reality games. *Journal of Virtual Reality and Broadcasting*, 4(5):1860–2037, 2007.
- [21] Mark Skwarek. CreatAR: Augmented Reality App. En *ACM SIGGRAPH 2013*

- Mobile*, SIGGRAPH '13, págs 21:1–21:1. ACM, Nueva York, NY, EE.UU., 2013. ISBN 978-1-4503-2341-3. doi:10.1145/2503512.2503539.
- [22] Ian Sommerville. *Ingeniería del software*. Pearson Educación, Barcelona, España, 2005.
- [23] L. Enrique Sucar. *Visión Computacional*. Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, México, 2005.
- [24] Chek Tien Tan y Donny Soh. Augmented Reality Games: A Review. *Proceedings of Gameon-Arabia, Eurosis*, 2010.
- [25] Daniel Wagner y Dieter Schmalstieg. Design Aspects of Handheld Augmented Reality Games. *Pervasive Gaming Applications*, 2:159–178, 2007.
- [26] Markus Weilguny. *Design Aspects in Augmented Reality Games*. Ph.D. thesis, Hagenberg, Mühlkreis, Austria, 2006.
- [27] Richard Wetzell, Rod McCall, Anne-Kathrin Braun y Wolfgang Broll. Guidelines for Designing Augmented Reality Games. En *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*, Future Play '08, págs 173–180. ACM, Nueva York, NY, EE.UU., 2008. ISBN 978-1-60558-218-4. doi:10.1145/1496984.1497013.
- [28] Stephen B. Wicker y Vijay K. Bhargava. *Reed-Solomon Codes and Their Applications*. John Wiley and Sons, Inc., Nueva York, EE.UU., 1999. ISBN 0-7803-5391-9.
- [29] Alper Yilmaz, Omar Javed y Mubarak Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4):15–36, 2006.

FICHA AUTOBIOGRÁFICA

Emmanuel Alejandro García Solís

Candidato para el grado de Ingeniero en Tecnología de Software

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

USO DE MARCADORES PARA LA SUPERPOSICIÓN
DE MODELOS TRIDIMENSIONALES EN IMÁGENES

Nací en Monterrey, Nuevo León, el día 21 de agosto del año 1992. En cuanto a mi educación, inicié mis estudios de bachillerato en el año 2007 en la Preparatoria 15 Unidad Madero, perteneciente a la Universidad Autónoma de Nuevo León. Para mis estudios de licenciatura ingresé a la carrera de Ingeniería en Tecnología de Software, formando parte de la primera generación de la entonces nueva carrera en la Facultad de Ingeniería Mecánica y Eléctrica perteneciente a la Universidad Autónoma de Nuevo León.