

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**ALGORITMO DE ESTIMACIÓN DE POSE ORIENTADO A LA
INICIALIZACIÓN DE UN SISTEMA DE REALIDAD AUMENTADA
BASADO EN MODELOS 3D APLICADO AL PATRIMONIO CULTURAL**

Tesis para obtener el título profesional de Ingeniero Electrónico

AUTOR:

Ricardo Moisés Rodríguez Oceda

ASESOR:

Benjamín Castañeda Aphan

Lima, Abril, 2022

Informe de Similitud

Yo, Benjamín Castañeda Aphan,

docente de la Facultad de Ciencias e Ingeniería de la Pontificia

Universidad Católica del Perú, asesor(a) de la tesis/el trabajo de investigación titulado

ALGORITMO DE ESTIMACIÓN DE POSE ORIENTADO A LA INICIALIZACIÓN DE UN SISTEMA DE REALIDAD AUMENTADA BASADO EN MODELOS 3D APLICADO AL PATRIMONIO CULTURAL,



del/de la autor(a)/ de los(as) autores(as)

Ricardo Moisés Rodríguez Oceda

dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 13%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 23/03/2023.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de Suficiencia Profesional, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha: Lima, 27/03/2023

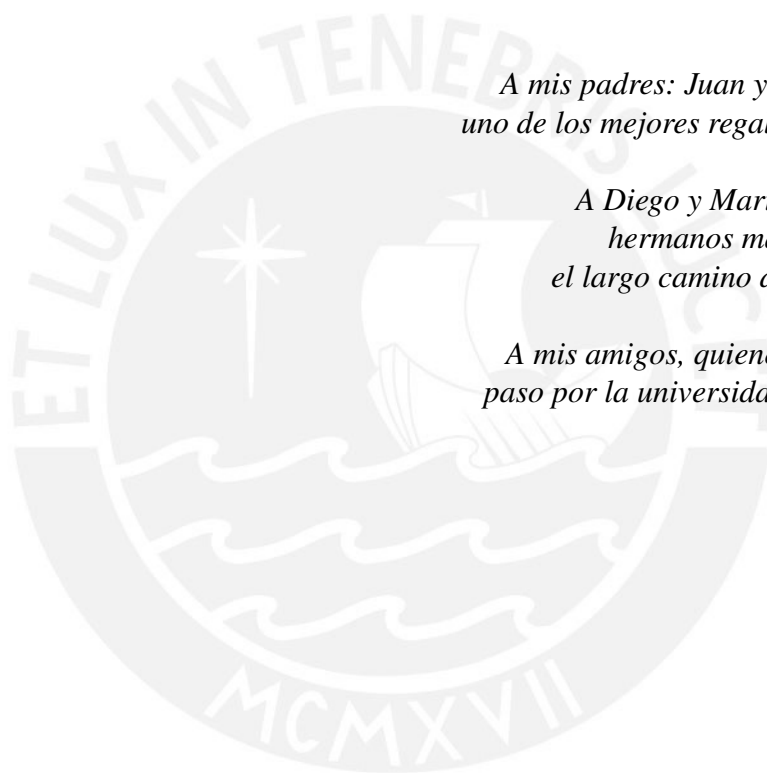
Apellidos y nombres del asesor / de la asesora: <u>Castañeda Aphan, Benjamín</u>	
DNI: 10791304	Firma 
ORCID: https://orcid.org/0000-0002-1913-0636 	

Resumen

La estimación de pose a partir de modelos 3D es un problema muy común dentro de las aplicaciones de robótica, tales como la realidad aumentada, la detección de objetos, el modelamiento 3D fotorrealista, entre otras. Dicha estimación consiste en la obtención de los parámetros extrínsecos de una cámara (posición y orientación) en un sistema de coordenadas determinado, a partir de una imagen capturada por dicha cámara, los parámetros intrínsecos de la misma y el modelo 3D del objeto o escena que se quiera detectar.

La realidad aumentada aplicada al patrimonio cultural pretende mejorar la experiencia de aprendizaje en lugares arqueológicos. En estos sistemas se emplea diferentes métodos para estimar la posición de la cámara; estos pueden ser basados en la detección de bordes, la detección de puntos característicos, entre otros. La elección del método a emplear depende de las características que posea el escenario a ser detectado.

En este trabajo se realizó un estudio de los principales métodos de estimación de pose basados en modelos 3D. Asimismo, se presenta la implementación y validación de un algoritmo de estimación de posición, orientado a la inicialización de un sistema de realidad aumentada basado en modelos 3D aplicado al patrimonio cultural, particularmente en este trabajo, la Huaca de la Luna. El desarrollo de este sistema presenta una metodología de diseño compuesta por diferentes bloques. En cada bloque se seleccionaron diferentes algoritmos, los cuales fueron evaluados tomando en consideración los valores de precisión y exactitud de los resultados de Rotación y Traslación, obtenidos por cada uno de ellos. De esta manera se llegó a una solución robusta y eficiente.



A mis padres: Juan y Rosa, por darme uno de los mejores regalos, la educación.

A Diego y Martín, quienes como hermanos mayores, allanaron el largo camino de la universidad.

A mis amigos, quienes hicieron que el paso por la universidad sea inolvidable.



Agradecimientos a mi asesor Dr. Benjamín Castañeda, por ser mentor en el camino de la investigación. Por la confianza y apoyo mostrado durante el desarrollo de esta tesis.

Asimismo, agradecimientos a CIENCIACTIVA por proveer fondos para desarrollar la investigación "Monitoreo remoto de la salud estructural de edificaciones emblemáticas de adobe: Integración de conocimiento y tecnología para un diagnóstico estructural adecuado" (PROYECTO ID 222-2015 FONDECYT) en el marco de la cual se ha ejecutado el presente trabajo.

TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO ELECTRÓNICO

Título : Algoritmo de estimación de pose orientado a la inicialización de un sistema de realidad aumentada basado en modelos 3D aplicado al patrimonio cultural

Área : Procesamiento Digital de Imágenes # 1341

Asesor : Dr. Benjamín Castañeda Aphan

Alumno : Ricardo Moisés Rodríguez Oceda

Código : 20110527

Fecha : 04/11/2016



Descripción y Objetivos

La estimación de pose (o de posición) a partir de modelos 3D es un problema muy común dentro de las aplicaciones de robótica, tales como la realidad aumentada, la detección de objetos, el modelamiento 3D fotorrealista, entre otras. Dicha estimación consiste en la obtención de los parámetros extrínsecos de una cámara (posición y orientación) en un sistema de coordenadas determinado, a partir de una imagen capturada por dicha cámara, los parámetros intrínsecos de la misma y el modelo 3D del objeto o escena que se quiera detectar. La realidad aumentada aplicada al patrimonio cultural pretende mejorar la experiencia de aprendizaje en lugares arqueológicos. En estos sistemas se emplea diferentes métodos para estimar la posición de la cámara; estos pueden ser basados en la detección de bordes, la detección de puntos característicos, entre otros. La elección del método a emplear depende de las características que posea el escenario a ser detectado. El objetivo principal de la presente tesis consiste en implementar y validar un algoritmo de estimación de posición, orientado a la inicialización de un sistema de realidad aumentada basado en modelos 3D aplicado al patrimonio cultural, específicamente, la Huaca de la Luna. Asimismo, se propone:

- Estudiar métodos eficientes para el desarrollo de algoritmos de estimación de pose a partir de modelos 3D.
- Implementar un algoritmo de estimación de pose sin marcadores basada en modelos 3D, empleando el software MATLAB.
- Obtener el modelo 3D del recinto esquinero de la Huaca de la Luna a partir de la técnica de fotogrametría.
- Validar la precisión del algoritmo implementado empleando el modelo 3D obtenido.



PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

M. Sc. Ing. MIGUEL ANGEL CATANO SÁNCHEZ
de la Especialidad de Ingeniería Electrónica

MÁXIMO 50 PÁGINAS

TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO ELECTRÓNICO

Título : Algoritmo de estimación de pose orientado a la inicialización de un sistema de realidad aumentada basado en modelos 3D aplicado al patrimonio cultural

Índice

Introducción

1. Estimación de pose en los sistemas de realidad aumentada.
2. Marco teórico: Realidad aumentada y geometría proyectiva.
3. Implementación del algoritmo de estimación de pose.
4. Resultados.

Conclusiones

Recomendaciones

Bibliografía

Anexos

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA


M. Sc. Ing. MIGUEL ÁNGEL CATANO SÁNCHEZ
Coordinador de la Especialidad de Ingeniería Electrónica



Índice general

Índice de figuras	iii
Índice de tablas	vi
Introducción	1
1. Estimación de pose en los sistemas de realidad aumentada	2
1.1. Descripción y formulación del problema	2
1.2. Estado del arte	3
1.3. Importancia y justificación del estudio	4
1.4. Objetivos	5
1.4.1. Objetivo general	5
1.4.2. Objetivos específicos	5
2. Marco teórico: Realidad Aumentada, Geometría Proyectiva y Detección de puntos característicos	6
2.1. Realidad Aumentada	6
2.1.1. Clasificación de los sistemas de RA	6
2.1.2. Métodos de estimación de pose	7
2.2. Geometría proyectiva	10
2.2.1. Puntos y transformaciones proyectivas	10
2.2.2. Geometría de la cámara	10
2.2.3. Calibración de la cámara	12
2.3. El problema Perspective n-Point	13
2.3.1. Transformación Linear Directa (TLD)	13
2.3.2. Tres Puntos en Perspectiva (P3P)	15
2.3.3. Perspective n-Point Eficiente (EPnP)	15
2.3.4. Perspective n-Point Robusto y Eficiente (REPPnP)	16
2.4. Detección de puntos característicos	16
2.4.1. Descriptores SIFT	16
2.4.2. Descriptores Surf:	17

3. Implementación del algoritmo de estimación de pose	18
3.1. Planteamiento General	18
3.1.1. Contexto	18
3.1.2. Requerimientos	18
3.1.3. Diseño	19
3.2. Etapas de la implementación del algoritmo de estimación de pose . . .	21
3.2.1. Modelamiento 3D (1)	25
3.2.2. Almacenamiento de puntos característicos 3D (2)	25
3.2.3. Adquisición y procesamiento de imágenes de entrada (3) . . .	27
3.2.4. Correspondencia de Puntos (4)	28
3.2.5. Estimación de pose (5)	28
3.2.6. Recursos empleados en la implementación	30
4. Experimentos y resultados	31
4.1. Método de Validación	31
4.1.1. Experimento 1: Empleando imágenes sintéticas	31
4.1.2. Experimento 2: Empleando imágenes reales	33
4.2. Resultados	34
4.2.1. Resultados del experimento 1	34
4.2.2. Resultados del experimento 2	37
4.2.3. Discusión de resultados	39
Conclusiones	40
Recomendaciones	41
Bibliografía	42

Índice de figuras

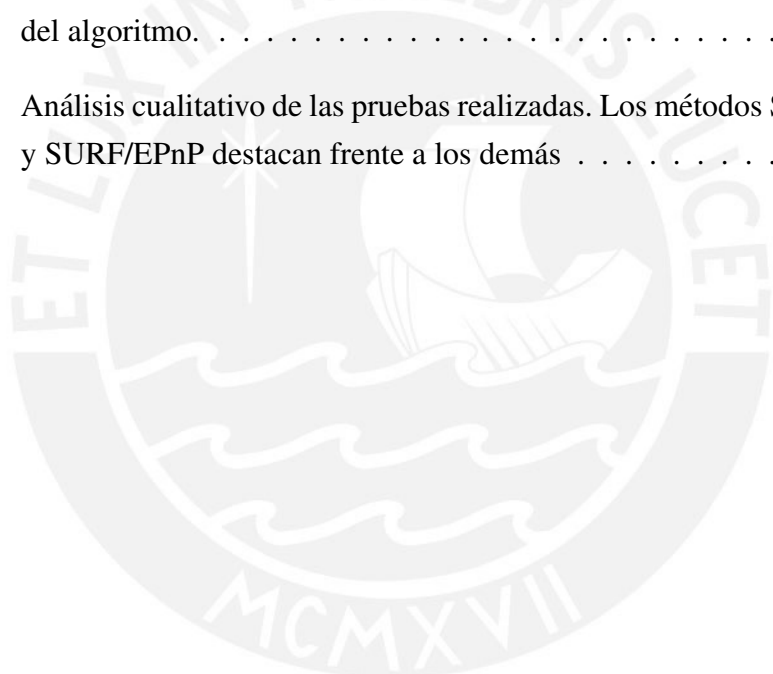
2.1. Tipos de Realidad Aumentada: (a) RA basada en marcadores; (b) RA ausente de marcadores. Imágenes obtenidas de la galería de imágenes de [11].	7
2.2. Taxonomía de sistemas de realidad aumentada basadas en modelos 3D. La estimación de pose puede clasificarse según su naturaleza de rastreo en rastreo recursivo y por detección. Imagen adaptada de [13].	8
2.3. Modelo de la cámara pinhole. C representa el centro de la cámara; X , un punto en el espacio y x , la proyección de dicho punto en el plano imagen. Imagen extraída de [24].	11
2.4. Imagen de un tablero de ajedrez, empleada durante el proceso de calibración de la cámara. Elaboración propia.	12
2.5. Calibración y corrección de distorsión de una cámara: En (a) se observa una imagen original capturada por una cámara, en líneas discontinuas se resalta la distorsión de las rectas; En (b) se presenta la imagen sin distorsión, luego de rectificarla empleando los coeficientes de distorsión. Imágenes tomadas de [23]	13
2.6. Problemática Perspective n-Point. Dadas las correspondencias entre puntos 3D en un sistema de coordenadas determinado con puntos 2D en el plano de imagen de una cámara, se debe encontrar los parámetros de rotación y traslación de dicha cámara. Imagen tomada de [22].	14
2.7. Representaciones del algoritmo SIFT, imágenes extraídas de [35]. En (a) se muestra la representación de la obtención de DoG. En (b) se observa una representación de los histogramas de una imagen, los cuales definen los descriptores SIFT.	17

3.1. Estudio del escenario a emplear. El recinto esquinero se encuentra ubicado en la parte baja de la Huaca de la Luna (a), los turistas tienen acceso a este lugar hasta determinada posición según como se observa en (b), imagen obtenida de [39]; desde tal posición el punto de vista de un visitante es el que se muestra en la Figura (c); finalmente en (d) se muestra la textura que presenta esta sección, ideal para realizar un reconocimiento de puntos característicos.	19
3.2. Diagrama de bloques del diseño del algoritmo propuesto. Imagen inspirada en [21]. Este diseño consta de dos etapas: una etapa de entrenamiento y una etapa de detección, la cual corresponde al proceso de inicialización del sistema de RA. Durante la primera etapa se realiza la reconstrucción del modelo 3D (1) y se almacenan los descriptores puntos característicos y sus correspondencias 3D (2). Estos descriptores son comparados con los descriptores obtenidos en la etapa de detección (3) y se consigue una correspondencia de puntos 2D con puntos 3D (4). Finalmente se emplea un algoritmo de estimación de pose PnP (5) para encontrar los parámetros de rotación y traslación.	22
3.3. Modelamiento 3D empleando el software Agisoft PhotoScan. En la Figura se observan las imágenes registradas para la reconstrucción del modelo así como las distancias empleadas para escalar el modelo 3D, 4 m (puntos 1 y 3), 0.9 m (puntos 1 y 2).	26
3.4. Ejemplo de correspondencia de puntos característicos empleando descriptores SURF. A la izquierda, imagen de entrenamiento. A la derecha, imagen de entrada de celular. En líneas amarillas se muestra las correspondencias obtenidas	29
3.5. Luego de emplear el algoritmo RANSAC implementado, se obtiene una estimación de posición a partir de un conjunto de correspondencias libre de outlier. De acuerdo a los colores, las líneas verdes representan los inliers; mientras que las líneas negras, los outliers. Las líneas azules representan la proyección de un cubo, que fue posicionado en ese lugar para realizar pruebas preliminares.	30
4.1. Diagrama de bloques de los métodos a evaluar SIFT/REPPnP, SIFT/EPnP, SURF/REPPnP y SURF/EPnP.	32
4.2. Generación de imágenes sintéticas empleando el software Blender. Se generaron 50 imágenes utilizando cámaras virtuales localizadas a 14 metros del recinto esquinero.	32

4.3. Error de Reproyección, se definen 11 puntos, los cuales servirán para determinar el error de reproyección medio de los algoritmos implementados. Las líneas negras representan los puntos de verificación; mientras que las líneas azules y amarillas representan los puntos de los algoritmos implementados.	33
4.4. Resultados de los métodos SIFT/REPPnP, SIFT/EPnP, SURF/REPPnP y SURF/EPnP para distintos números de correspondencias. (a) Error de traslación. (b) Error de rotación. (c) Porcentaje de estimaciones exitosas. (d) Error máximo de reproyección.	35
4.5. Resultados de los métodos SIFT/REPPnP, SIFT/EPnP, SURF/REPPnP y SURF/EPnP empleando 110 puntos de correspondencia 2D - 3D. (a) Error de traslación. (b) Error de rotación. (c) Porcentaje de estimaciones exitosas. (d) Error máximo de reproyección.	36
4.6. Resultados de la tasa de estimaciones exitosas para los métodos SIFT/REPPnP, SIFT/EPnP, SURF/REPPnP y SURF/REPPnP frente a imágenes con cambios de luminosidad	37
4.7. Pruebas de confiabilidad. Las imágenes (a),(c) y (e) son una muestra de las fotografías que se capturan durante la mañana; por otra parte, las imágenes (b), (d) y (f) fotografías capturadas por la tarde. Se determinó visualmente la cantidad de aciertos de la estimación de pose.	38

Índice de tablas

3.1. Análisis de métodos de estimación de pose. Métodos evaluados basados en: marcadores retro-reflectivos, marcadores visuales, puntos de interés 2D, puntos de interés 3D y bordes 3D. Comparación obtenida de [18] y [13]	20
3.2. Descripción de los recursos empleados por cada etapa en el desarrollo del algoritmo.	30
4.1. Análisis cualitativo de las pruebas realizadas. Los métodos SIFT/EPnP y SURF/EPnP destacan frente a los demás	39



Introducción

La estimación de la posición de una cámara a partir de modelos 3D es un problema muy común en varias aplicaciones dentro del área de visión por computadora, tales como la realidad aumentada, la detección de objetos, el modelamiento 3D fotorrealista, entre otras aplicaciones. Dicha estimación consiste en la obtención de los parámetros extrínsecos de una cámara (posición y orientación) en un sistema de coordenadas determinado, a partir de una imagen capturada por dicha cámara, los parámetros intrínsecos de la misma y el modelo 3D del objeto o escena que se quiera detectar.

Actualmente, existen distintos planteamientos que solucionan esta problemática, estos pueden estar basados en la detección de bordes o de puntos característicos, por ejemplo. La elección del tipo de solución a implementar dependerá del tipo de aplicaciones que se quiera desarrollar.

En el caso de las aplicaciones de Realidad Aumentada basadas en el patrimonio cultural, se deben tener en consideración factores como la textura del modelo 3D, o si la escena que se tratará posee obstáculos que puedan interferir en la estimación de la posición, por ejemplo.

Es debido a ello, que en la presente tesis se realiza un estudio de un método de estimación de pose basado en puntos característicos aplicado a un monumento del patrimonio cultural peruano; se busca evaluar la confiabilidad del algoritmo así como su precisión. El escenario a estudiar será el "recinto esquinero" dentro de la Huaca de La Luna, ubicada en el departamento de La Libertad.

El desarrollo de esta tesis está organizado de la siguiente manera: En el **Capítulo 1** se definirá la problemática detalladamente, se expondrán los recientes estudios con respecto a las técnicas empleadas en los sistemas de realidad aumentada aplicados al patrimonio cultural y se plantearán los objetivos del presente trabajo. En el **Capítulo 2** se estudiarán los conceptos teóricos que se utilizaron para el desarrollo de esta tesis, se realiza una visión general de los sistemas de realidad aumentada y de los principales conceptos de geometría proyectiva. En el **Capítulo 3** se desarrolla el método a implementar y se describe cada etapa detalladamente. Finalmente, en el **Capítulo 4** se realiza la evaluación del algoritmo implementado. En este capítulo se analiza la precisión y la exactitud del método implementado; seguido de las **Conclusiones y Recomendaciones**.

Capítulo 1

Estimación de pose en los sistemas de realidad aumentada

En el siguiente capítulo se planteará la problemática de la presente tesis, se presentarán los trabajos realizados en la literatura científica que resuelven dicha problemática y finalmente, se describirán los objetivos del presente estudio.

1.1. Descripción y formulación del problema

En la actualidad, la realidad aumentada es un tema que llama la atención de muchos investigadores; esto es debido a las múltiples aplicaciones que ofrecen estos sistemas, tales como ser herramientas de educación, herramientas para la medicina, patrimonio cultural, entretenimiento, entre otras aplicaciones [1].

Por definición, un sistema de realidad aumentada (RA) es un sistema que complementa el mundo real con objetos virtuales (generados por computadora), capaz de generar la ilusión de coexistencia de ambos elementos en un mismo espacio. Para que este sistema cumpla su función de unir ambos ambientes: virtual y real, es necesario que posea una exactitud suficiente, es decir, que el objeto virtual sea posicionado en el lugar correcto y en el instante correcto.

Determinar la posición precisa del objeto virtual, o estimar su pose 3D, es un problema que ha sido bastante estudiado en los últimos años, dado que esto representa la base de múltiples aplicaciones de visión por computadora, como por ejemplo los sistemas de localización de robots, sistemas de vigilancia o rastreo de objetos. Diferentes métodos han sido propuestos para resolver dicha problemática, la selección de estos depende del tipo de aplicación que se requiera; como se explicará más adelante, la presente tesis está enfocada en aplicaciones orientadas al patrimonio cultural.

Los sistemas de realidad aumentada, al igual que los de rastreo de objetos, constan de dos etapas: Inicialización y seguimiento. La diferencia entre ambas etapas consiste

en que durante la etapa de inicialización no se tiene conocimiento alguno de la posición aproximada del objeto, mientras que en la etapa de seguimiento se tiene como referencia la posición estimada en el cuadro anterior (asumiendo que se trabaja con una secuencia de imágenes). Con la información del cuadro anterior es posible reducir los errores de precisión.

La etapa de inicialización, al no tener referencia previa de la posición del objeto, representa un problema puesto que se deben desarrollar técnicas que permitan estimar la posición exacta del objeto o escenario en el menor tiempo posible.

La problemática que se aborda en la presente tesis consiste en la obtención automática de la pose 3D de una cámara (orientación y posición) en un sistema de coordenadas determinado, a partir de una imagen de entrada, los parámetros intrínsecos de la cámara que capturó la imagen y el modelo 3D del escenario capturado. En otras palabras se busca realizar el registro de una imagen 2D a partir de un modelo 3D.

1.2. Estado del arte

La presente tesis forma parte del proyecto de Realidad Aumentada del Laboratorio Engineering and Heritage de la PUCP. Este proyecto consiste en la implementación de un sistema de RA aplicado en monumentos arqueológicos, en el cual se emplea un dispositivo móvil como una ventana y hace posible visualizar dicho monumento arqueológico reconstruido; para este caso de estudio, se eligió a la Huaca de la Luna como escenario.

A continuación, se realizará un breve recuento de los trabajos realizados en el campo de realidad aumentada aplicados al patrimonio cultural, en el cual se dará énfasis a la etapa de inicialización.

El estudio de sistemas de RA aplicados al patrimonio cultural ha sido desarrollado por muchos investigadores durante estos últimos años. En [2] se puede encontrar un resumen de varios de estos proyectos. Recientemente, se ha realizado un proyecto de RA en el cual se emplea a las ruinas de Chan Chan como escenario [3]. Este sistema plantea una solución basada en la localización, empleando el GPS y el giroscopio de los equipos móviles (tablets y celulares) y el kit de desarrollo de software (SDK) Wikitude [4]. Por otro lado, en [5] se plantean sistemas de RA para ser usadas como guías dentro museos. Los métodos planteados en este caso son basados en imágenes y puntos característicos. En [6] se realiza un sistema basado en imágenes, empleando la aplicación Augment para poder visualizar las murallas Aurelianas de Roma reconstruidas in situ. El primer paso para la creación de este sistema es la reconstrucción 3D del muro. Luego, se crea un conjunto de datos de fotos capturadas desde diferentes puntos de vista. Cada una de estas fotos debe cargarse al sitio web

de Augment donde se convertirán en marcadores de referencia. Finalmente, para cada foto se debe alinear la pose del modelo 3D correspondiente. De esta manera, cuando algún usuario capture una imagen de alguna muralla desde su celular, la aplicación se encargará de elegir la foto dentro del dataset que posea mayor similitud con la imagen capturada y calculará la posición del modelo 3D por medio de homografía. Por otra parte, en [7] se diseña un sistema de RA para ser aplicado en recorridos turísticos. Se plantea emplear técnicas similares a las mencionadas anteriormente, así como métodos de detección de objetos 3D.

Según lo mencionado hasta el momento, diversos métodos pueden ser aplicados para detectar la posición de un objeto. En la siguiente tesis se estudiará el método de estimación de pose basado en modelos 3D.

1.3. Importancia y justificación del estudio

Esta tesis busca resolver el problema de inicialización de un sistema de RA aplicado al patrimonio cultural. El estudio de la exactitud de estimación de pose de una cámara en la etapa de inicialización de un sistema de RA es de vital importancia pues a partir de esta primera aproximación, seguirán las posteriores estimaciones durante la etapa de seguimiento. Asimismo, permitirá validar las características de determinados objetos arqueológicos (Huacas) como escenario válido para el planteamiento propuesto.

La exactitud estudiada garantizará el correcto desempeño de un sistema de RA, dado que este requiere de una alta exactitud para poder crear la ilusión de un escenario real combinado con objetos virtuales.

Este estudio servirá para la realización de futuros trabajos en la implementación del sistema de RA completo. Se podrá comparar distintas propuestas de métodos durante la implementación del sistema de RA, de modo que se pueda elegir la mejor alternativa. Asimismo, permitirá evaluar el potencial de las aplicaciones de RA aplicadas al patrimonio cultural, para posteriormente adquirir algún SDK comercial, como los mencionados anteriormente.

De manera general, los sistemas de realidad aumentada aplicada en patrimonio cultural son empleados no solo para los turistas como atractivo turístico, sino también son una herramienta empleada por los arqueólogos; la cual permitirá un mejor entendimiento de la historia del lugar.

1.4. Objetivos

1.4.1. Objetivo general

Implementar y validar un algoritmo de estimación de posición, orientado a la inicialización de un sistema de realidad aumentada basado en modelos 3D aplicado al patrimonio cultural, específicamente, la Huaca de la Luna.

1.4.2. Objetivos específicos

- Estudiar métodos eficientes para el desarrollo de algoritmos de estimación de pose a partir de modelos 3D.
- Implementar un algoritmo de estimación de pose sin marcadores basada en modelos 3D.
- Obtener el modelo 3D del recinto esquinero de la Huaca de la Luna a partir de la técnica de fotogrametría.
- Validar la exactitud del algoritmo implementado empleando el modelo 3D obtenido.

Capítulo 2

Marco teórico: Realidad Aumentada, Geometría Proyectiva y Detección de puntos característicos

En este capítulo se presentará una clasificación de los sistemas de RA, así como los conceptos matemáticos elementales para el entendimiento del algoritmo a desarrollar.

2.1. Realidad Aumentada

La definición de RA fue mencionada en el capítulo anterior, a continuación se detallan los tres criterios fundamentales que debe cumplir todo sistema de RA [8]: En primer lugar, un sistema de RA debe combinar ambos ambientes, virtual y real en un mismo espacio. En segundo lugar, este debe ser interactivo y en tiempo real. Finalmente, debe permitir realizar un correcto registro entre objetos reales y virtuales.

2.1.1. Clasificación de los sistemas de RA

Hasta el momento se han presentado distintos métodos empleados en los sistemas de RA. A continuación, se presentará una clasificación de estos sistemas para poder ubicarnos dentro del desarrollo de los mismos. Esta clasificación comprende solo el estudio de métodos visuales, por lo cual no se considerarán los sistemas basados en GPS dado que estos no están basados en técnicas procesamiento de imágenes.

Los sistemas de RA pueden ser clasificados en dos tipos (Ver Figura 2.1). El más popular de ellos es el **sistema basado en marcadores**. Este sistema consiste en el empleo de marcadores artificiales como puntos de referencia para la cámara, dichos marcadores se posicionan en el objeto o en medio de la escena a detectar. Esta alternativa es una solución práctica y de fácil implementación, dado que permite estimar la posición de manera precisa y eficiente (ver Figura 2.1(a)); sin embargo, existen situaciones en las que el escenario no permite la colocación de un marcador y es

necesario emplear sus características naturales como referencia. Un ejemplo de estos casos son los sistemas de RA aplicados en la arqueología ya que se busca preservar la estructura del escenario original. A estos sistemas se les conoce como **sistema ausente de marcadores** y su tiempo de procesamiento para este tipo de sistemas es superior. Actualmente, desarrolladores como ARmedia [9], Metaio [10] o Wikitude [4] se encuentran trabajando en herramientas de desarrollo para implementar estos sistemas.

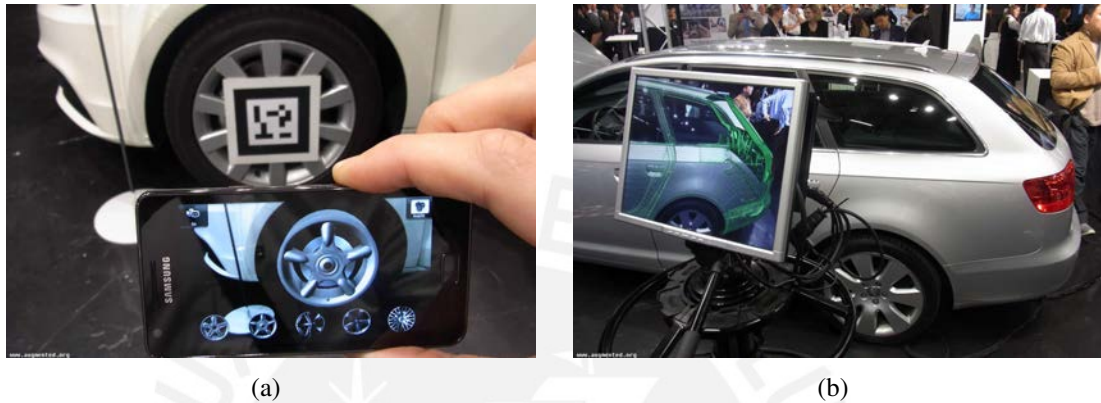


Figura 2.1: Tipos de Realidad Aumentada: (a) RA basada en marcadores; (b) RA ausente de marcadores. Imágenes obtenidas de la galería de imágenes de [11].

Los sistemas ausentes de marcadores generalmente se basan en **imágenes** mediante el rastreo de puntos característicos. No obstante, este tipo de planteamiento resulta limitado cuando se trabaja con un escenario de geometría compleja o aquellos en los que el objeto o escenario a detectar no cuenta con texturas. Debido a esto, se propone un nuevo planteamiento, estos son los **sistemas ausentes de marcadores basados en el rastreo 3D**, en los cuales se emplea un modelo 3D como referencia.

Este último tipo de sistema mencionado puede ser clasificado a su vez en dos tipos: aquellos que son **basados en modelos 3D** y **los que se basan en la estructura obtenida a partir del movimiento** (o SFM por sus siglas en inglés), según como se describe en [12]. La diferencia entre ambos consiste en que mientras que en los sistemas basados en modelos 3D se tiene información a priori del espacio real, en los sistemas basados en SFM esta información es obtenida a medida que se realiza el rastreo.

2.1.2. Métodos de estimación de pose

Dentro del contexto de los sistemas de RA basados en modelos 3D se han desarrollado distintas técnicas que permiten la estimación de pose de la cámara. Dichas técnicas pueden ser clasificadas según su naturaleza de rastreo (ver Figura 2.2), esto es: **rastreo recursivo** y **rastreo por detección** [13]. En el rastreo recursivo se emplea la

pose estimada en un cuadro anterior como referencia para poder calcular la pose actual, mientras que en el rastreo por detección la posición es estimada sin alguna información previa.

Si bien, en ambos métodos se emplean las características naturales de la escena, cabe resaltar que existen distintas características a tomar en cuenta. Estas pueden ser **bordes**, variaciones de puntos obtenidas a partir de la estimación del **flujo óptico** "optical flow" o del análisis de la **textura** propia del escenario u objeto a detectar, tal como se observa en la Figura 2.2.

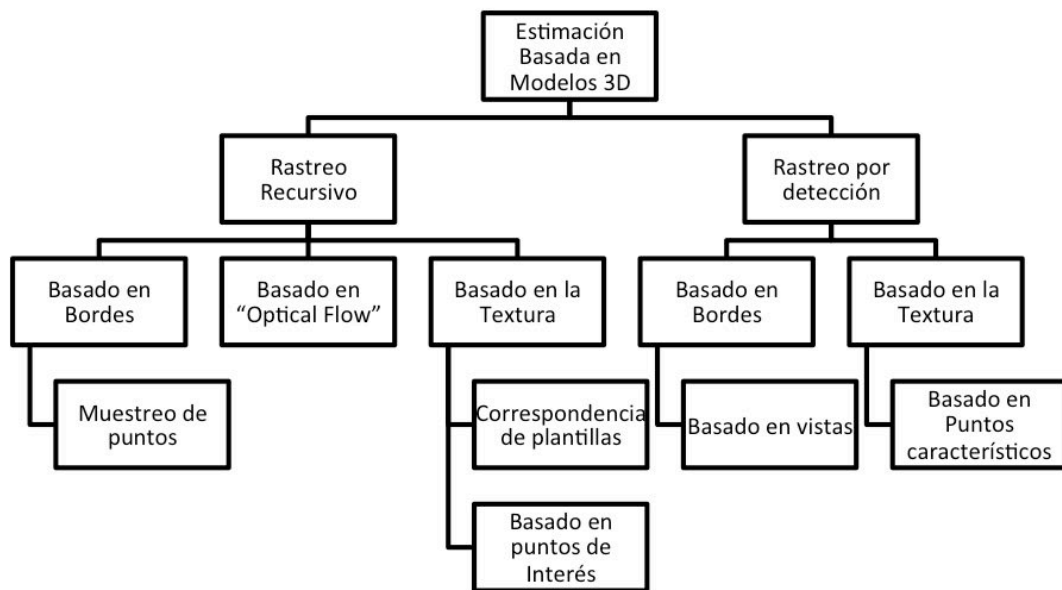


Figura 2.2: Taxonomía de sistemas de realidad aumentada basadas en modelos 3D. La estimación de pose puede clasificarse según su naturaleza de rastreo en rastreo recursivo y por detección. Imagen adaptada de [13].

Dentro de la clasificación de rastreo recursivo, el método basado en bordes [14] plantea una solución a partir del **muestreo de puntos**. Esto consiste en rastrear puntos de control obtenidos del muestreo de los bordes del modelo 3D y compararlos con los puntos obtenidos en la imagen empleando la misma estrategia. El método basado en **flujo óptico** [15] plantea realizar un seguimiento a los puntos que varían en el tiempo dentro de una secuencia de imágenes, analizando magnitud y dirección; de tal modo que, se pueda proyectar este desplazamiento de puntos 2D a puntos 3D y obtener la variación de la posición. Por último, en el método basado en la textura se plantean dos tipos de soluciones. La primera de ellas se basa en la **correspondencia de plantillas** [16] en donde se fija una posición inicial de un modelo 3D con respecto a una imagen inicial tomada como plantilla. La estimación de la posición del modelo en las subsecuentes imágenes dependerá de las variaciones de niveles de gris entre la plantilla y la región predicha de la imagen. La segunda solución está basada en la

detección de **puntos de interés** [17]. En esta se propone rastrear imágenes por medio de puntos como esquinas, por ejemplo. Para ello se extraen distintas vistas del modelo 3D y durante el rastreo se genera una vista intermedia por medio de homografía. Con ello es posible estimar el movimiento del objeto mediante la comparación de dichos puntos de interés.

Con respecto al rastreo por detección, se plantea un método basado en bordes y otro basado en la textura. Estos métodos serán explicados con más detalle en las siguientes líneas, dado que este tipo de planteamientos permiten resolver la etapa de inicialización de los sistemas de RA, el cual es el objetivo principal de esta tesis. Para mayor información de los métodos, el lector puede referirse a [18].

Rastreo por detección basado en bordes

Esta técnica ha sido planteada en [19]. Se dice que está basada en vistas debido a que en una etapa anterior se obtienen imágenes del objeto a detectar desde distintas posiciones. Se crea un modelo 2D de cada imagen, tomando en consideración la magnitud y dirección de sus bordes detectados. Luego, se recibe una imagen de entrada y se comparan las medidas de similitud realizando un producto punto entre las gradientes de las direcciones de los bordes correspondientes entre el modelo y dicha imagen de entrada. Finalmente, es posible obtener la vista más cercana y se emplean métodos de rastreo recursivo para obtener la posición del objeto. Este tipo de técnica resulta útil para modelos 3D que no presentan textura como es el caso de las aplicaciones de RA en ambientes industriales.

Rastreo por detección basado en textura

La presente técnica está basada en la metodología propuesta en [20]. Esta técnica es similar a la de los puntos de interés, mencionada anteriormente; sin embargo, esta se basa en obtener la pose a partir de la correspondencia de puntos 2D - 3D. Para ello, se plantean dos etapas: La primera de ellas es la etapa de entrenamiento en la cual se capturan distintas vistas del modelo 3D y se obtienen descriptores invariantes en el tiempo y en escala, las proyecciones 3D de los puntos 2D son obtenidas y almacenadas en memoria. La siguiente etapa es la de ejecución, en esta se recibe una imagen de entrada, se extraen sus descriptores y se comparan con los descriptores almacenados en memoria. De esta manera se consigue establecer las correspondencias de puntos 2D con puntos 3D, las cuales son empleadas para obtener la estimación de la pose. Esta técnica ha sido empleada en distintos proyectos, tales como [21], [22]; asimismo, es la base de varios algoritmos de detección facial.

2.2. Geometría proyectiva

En esta sección se estudiarán los conceptos matemáticos principales para el desarrollo de esta tesis. Más información sobre el desarrollo teórico puede ser encontrados en [23] y [18].

2.2.1. Puntos y transformaciones proyectivas

Un punto en el espacio euclidiano R^3 es representado por tres coordenadas (x, y, z) . En geometría proyectiva, este punto es representado por un vector de cuatro dimensiones llamado **vector homogéneo** $X = (x_1, x_2, x_3, x_4)^T$ con $x_4 \neq 0$ donde $x = x_1/x_4$, $y = x_2/x_4$ y $z = x_3/x_4$.

En un espacio P^3 , una transformación proyectiva consiste en la transformación lineal de un vector homogéneo y se representa por una matriz 4x4 no singular: $X' = HX$. La matriz H representa la transformación proyectiva y se le conoce como **matriz homogénea**.

2.2.2. Geometría de la cámara

El funcionamiento de una cámara consiste en mapear puntos de un espacio 3D dentro de una imagen plana 2D. Los modelos de cámaras son empleados para entender este funcionamiento de manera analítica; el modelo de cámara más simple es el de la **cámara pinhole**. Bajo este modelo, un punto 3D X_{world} en el espacio es proyectado en un punto 2D x_{cam} , el cual viene a ser el punto de intersección entre la línea que une X_{world} y el centro de la cámara C , con el plano de la imagen, según como se observa en la Figura 2.3.

A este tipo de proyección se le conoce como **proyección perspectiva**. Cuando X_{world} y x_{cam} están expresadas en coordenadas homogéneas ($X_{world} = [X, Y, Z, 1]$ y $x_{cam} = [u, v, 1]$) puede ser representada como

$$x_{cam} = PX_{world} \quad (2.1)$$

donde P es una matriz de 3x4, la cual representa la **matriz de proyección**. Dentro de esta matriz se encuentran los parámetros intrínsecos K y extrínsecos $[R|t]$ de la cámara.

$$P = K[R|t] \quad (2.2)$$

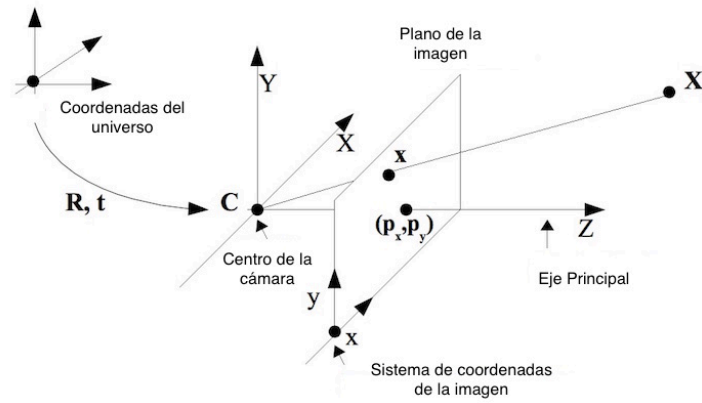


Figura 2.3: Modelo de la cámara pinhole. C representa el centro de la cámara; X, un punto en el espacio y x, la proyección de dicho punto en el plano imagen. Imagen extraída de [24].

Parámetros intrínsecos

Los parámetros intrínsecos o parámetros de calibración definen la geometría interna de la cámara, y están representados por

$$K = \begin{bmatrix} \alpha_u & s & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

donde $\alpha_u = fm_u$ y $\alpha_v = fm_v$ representan la distancia focal de la cámara en dimensiones de píxel, siendo m_u y m_v los factores de escalamiento equivalentes al número de píxeles por unidad en el plano real en las direcciones x e y . Asimismo, $\tilde{x}_0 = [u_0, v_0]^T$ es el centro del plano de la imagen en dimensiones de píxel, donde $u_0 = m_u p_u$ y $v_0 = m_v p_v$, siendo p_u y p_v el centro del plano en dimensiones reales. Finalmente, s es el valor skew. Este suele ser cero para la mayoría de las cámaras y define la perpendicularidad de las direcciones x e y .

Parámetros extrínsecos

Estos parámetros describen la relación que existe entre los sistemas de referencia del mundo real y el de la cámara. Es decir, describen la **pose de la cámara**, su orientación y posición. Estos parámetros son definidos por la matriz $[R|t]$ la cual es

la concatenación de la matriz de rotación y el vector de traslación, representada como

$$[R|t] = \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_1 \\ R_{21} & R_{22} & R_{23} & t_2 \\ R_{31} & R_{32} & R_{33} & t_3 \end{bmatrix} \quad (2.4)$$

La matriz de rotación además de ser representada como matriz, también puede ser representada por otros tipos de parametrización como **ángulos de euler**, **cuaterniones** o **mapas exponenciales** [18].

2.2.3. Calibración de la cámara

En la mayoría de algoritmos de visión por computadora es necesario tener conocimiento de los parámetros intrínsecos de la cámara con la que se trabaja.

Existen muchos algoritmos que permiten conocer estos parámetros, a través de una etapa llamada **calibración de la cámara**.

Uno de los algoritmos más populares se basa en capturar múltiples imágenes de un tablero de calibración el cuál suele ser una imagen de un tablero de ajedrez, como se ve en la Figura 2.4. El objetivo es encontrar las correspondencias 2D - 3D entre los puntos detectados en la imagen (esquinas de cuadrados negros) y los puntos 3D reales, que se definen como un conjunto de puntos coplanares en el espacio, con ello es posible calcular la matriz de proyección. Este método ha sido planteado en [25], y herramientas como la aplicación "Camera calibrator" del toolbox "Sistemas de visión por computadora" de Matlab [26] facilitan este trabajo.



Figura 2.4: Imagen de un tablero de ajedrez, empleada durante el proceso de calibración de la cámara. Elaboración propia.

Empleando este método también es posible obtener los coeficientes de distorsión de la cámara. Esta distorsión es generada debido al tipo de lente que emplea una cámara. En la Figura 2.5, se pueden observar el efecto de la corrección de la distorsión. Mientras que en la Figura 2.5(a) las líneas punteadas siguen un camino curvo, en la Figura 2.5(b) la imagen se encuentra rectificadas y aquellas líneas se ven rectas.



Figura 2.5: Calibración y corrección de distorsión de una cámara: En (a) se observa una imagen original capturada por una cámara, en líneas discontinuas se resalta la distorsión de las rectas; En (b) se presenta la imagen sin distorsión, luego de rectificarla empleando los coeficientes de distorsión. Imágenes tomadas de [23]

2.3. El problema Perspective n-Point

El problema de **Perspective n-Point**, o también llamado estimación de pose a través de n puntos, nace a partir de la necesidad de calibrar una cámara [27]. El principal objetivo de este planteamiento consiste en obtener la posición y orientación de una cámara a partir de n correspondencias de puntos 3D con sus proyecciones en 2D.

Para las posteriores explicaciones, se asumirá n como el número de correspondencias entre puntos 3D M_i , y sus proyecciones 2D m_i (Ver Figura 2.6). Se sabe además que la relación entre estos puntos está dada por la matriz \mathbf{P} que proyecta M_i en m_i , y contiene los parámetros extrínsecos e intrínsecos de la cámara. En otras palabras, se determinará la relación $\mathbf{P}\tilde{M}_i \equiv \tilde{m}_i$ para todo i , donde \equiv representa la equivalencia, tomando en cuenta el factor de escalamiento, dado que se está trabajando con **coordenadas homogéneas**.

En las siguientes líneas se presentará las principales técnicas empleadas para obtener los parámetros de una cámara a través de la correspondencia entre puntos 2D con puntos 3D. Entre ellos se encuentran los algoritmos más básicos como P3P [28] y P4P [29]; y algoritmos más sofisticados como el algoritmo UPnP [30], EPnP [31] y REPnP [32], entre otros.

2.3.1. Transformación Linear Directa (TLD)

El presente algoritmo busca estimar los 11 parámetros de la matriz \mathbf{P} de la ecuación 2.2 (6 parámetros extrínsecos y 5 parámetros intrínsecos). La correspondencia entre

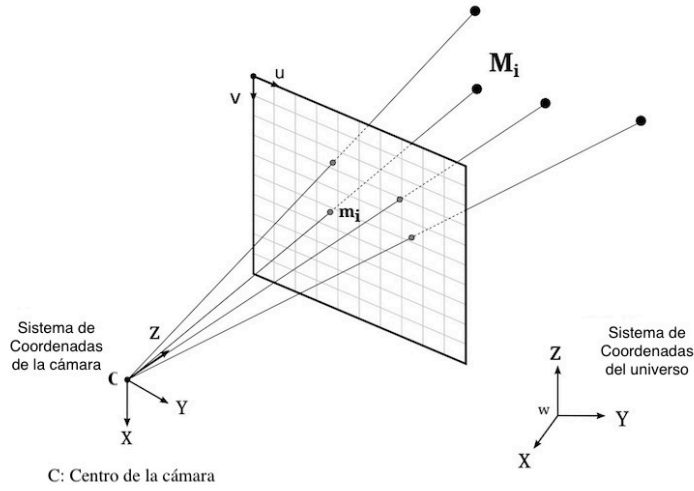


Figura 2.6: Problemática Perspective n-Point. Dadas las correspondencias entre puntos 3D en un sistema de coordenadas determinado con puntos 2D en el plano de imagen de una cámara, se debe encontrar los parámetros de rotación y traslación de dicha cámara. Imagen tomada de [22].

puntos 3D (X_i, Y_i, Z_i) y puntos 2D (u_i, v_i) puede ser expresada de la siguiente manera:

$$\frac{P_{11}X_i + P_{12}Y_i + P_{13}Z_i + P_{14}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}} = u_i, \quad (2.5)$$

$$\frac{P_{21}X_i + P_{22}Y_i + P_{23}Z_i + P_{24}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}} = v_i$$

Como se puede observar, por cada correspondencia entre punto 2D con 3D se tendrá dos ecuaciones. Dado que se quiere resolver 11 parámetros, el número mínimo de correspondencias para estimar la matriz P es de 6.

Este sistema de ecuaciones podría reordenarse de la siguiente manera:

$$P_{31}X_i u_i + P_{32}Y_i u_i + P_{33}Z_i u_i + P_{34}u_i - P_{11}X_i - P_{12}Y_i - P_{13}Z_i - P_{14} = 0, \quad (2.6)$$

$$P_{31}X_i v_i + P_{32}Y_i v_i + P_{33}Z_i v_i + P_{34}v_i - P_{21}X_i - P_{22}Y_i - P_{23}Z_i - P_{24} = 0$$

y viéndolo de manera matricial podría reescribirse de la siguiente manera:

$$A_i = \begin{bmatrix} -X_i & -Y_i & -Z_i & -1 & 0 & 0 & 0 & 0 & X_i u_i & Y_i u_i & Z_i u_i & u_i \\ 0 & 0 & 0 & 0 & -X_i & -Y_i & -Z_i & -1 & X_i v_i & Y_i v_i & Z_i v_i & v_i \end{bmatrix} \quad (2.7)$$

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} & P_{21} & P_{22} & P_{23} & P_{24} & P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix}^T$$

Finalmente, considerando que se tendrán 6 sistemas de ecuaciones, dados los 6 puntos a considerar, este sistema de ecuaciones puede expresarse como: $\mathbf{AP} = \mathbf{0}$, donde \mathbf{P} es un vector compuesto por los coeficientes P_{ij} . Este planteamiento permite desarrollar la ecuación aplicando la Descomposición de Valores Singulares (SVD, por sus siglas en inglés) de \mathbf{A} . Más detalles de este desarrollo pueden encontrarse en [23].

A pesar de que este método permite el cálculo de los parámetros extrínsecos e intrínsecos de la cámara, además de permitir realizar un cálculo directo sin iteraciones, no es recomendable emplearlo para el desarrollo de algoritmos de estimación de pose o rastreo, ya que no se podría garantizar que el resultado de los parámetros intrínsecos sea el mismo para cada estimación. En la calibración de las cámaras, por otro lado, sí se suele emplear este método dado que es posible obtener una media a partir de los distintos resultados obtenidos.

Debido a esto, se propone emplear este método para la etapa de calibración, y con los resultados obtenidos se procede a calcular únicamente los parámetros extrínsecos.

2.3.2. Tres Puntos en Perspectiva (P3P)

A diferencia del método TLD, en este planteamiento se asume el conocimiento de los parámetros intrínsecos de la cámara y se enfoca en la obtención de los parámetros extrínsecos. Como se puede suponer a partir de su nombre, este algoritmo se basa en la obtención de la posición y orientación de la cámara a partir de tres puntos correspondientes. Existen diferentes aproximaciones que plantean soluciones a este problema específico [27].

La idea básica de este planteamiento consiste en obtener inicialmente las distancias entre el centro de la cámara \mathbf{C} y los puntos 3D \mathbf{M}_i (Ver la Figura 2.6). Luego de ello, los puntos \mathbf{M}_i son expresados en las coordenadas de la cámara, \mathbf{M}_i^c . Posteriormente, se realiza el cálculo de la matriz $[R|t]$, empleando herramientas como cuaterniones o SVD, donde se busca alinear los puntos 3D del espacio con los puntos 3D del sistema de coordenadas de la cámara. Esta técnica de estimación da como resultado cuatro soluciones posibles las cuáles deberán ser evaluadas en una etapa de refinamiento.

2.3.3. Perspective n-Point Eficiente (EPnP)

Según lo descrito por los autores en [31] y lo presentado en [33], este método es una de las primeras estimaciones PnP de complejidad lineal $O(n)$ el cual emplea un método de iteración veloz que permite mejorar la precisión. La eficiencia proviene de la reducción del problema PnP a encontrar la posición de cuatro puntos de control que son una suma ponderada de todos los puntos 3D. Tras obtener una solución lineal, los pesos de los cuatro puntos de control se refinan mediante una optimización

de Gauss-Newton. Este algoritmo está diseñado para emplear múltiples puntos de correspondencia. Cuantos más puntos se utilicen, mayor será la precisión de la pose; a diferencia de muchos otros algoritmos PnP que están diseñados para n específicos (por ejemplo, P3P). Por otra parte, este algoritmo, como la mayoría de los algoritmos PnP, no está adaptado para los valores atípicos o outliers. De modo que, si los datos de entrada están contaminados con coincidencias 2D-3D incorrectas, se deberá emplear un algoritmo de detección de outliers como RANSAC [34], por ejemplo.

2.3.4. Perspective n-Point Robusto y Eficiente (REPPnP)

El método Robust Efficient Procrustes PnP, a diferencia del método anterior (EPnP), realiza los cálculos de detección de outliers y estimación de pose de manera simultánea. Esto implica que no es necesario un algoritmo RANSAC para determinar inliers y outliers, lo cual reduce el tiempo de estimación y brinda mayor precisión en los cálculos. Mientras que en el algoritmo EPnP, se realiza una aproximación geométrica, basado en la disminución de errores por reproyección; en el algoritmo REPPnP se reduce este error de manera algebraica.

2.4. Detección de puntos característicos

En el punto anterior se habló sobre la detección de pose a partir de puntos de correspondencia. El primer paso para poder realizar esta detección es la generación de puntos característicos. Existen diferentes algoritmos que permiten extraer puntos característicos, tales como SIFT [35], SURF[36], FAST [37] u ORB [38]. A continuación se detallarán los algoritmos SIFT y SURF, los cuales serán empleados más adelante en esta tesis.

2.4.1. Descriptores SIFT

El algoritmo SIFT (*Scale Invariant Feature Transform*) se emplea para extraer características invariantes en escala y en rotación. Inicialmente, se realiza la **detección de características** invariantes en escala y rotación. Para ello la imagen de entrada es llevada a diferentes escalas. A cada una de las imágenes escaladas se le aplica filtros gaussianos con coeficientes diferentes. De este modo, se obtiene un espacio de escalas gaussianas. Finalmente, se realiza una operación de diferencia entre cada una de las imágenes obtenidas (ver Figura 2.7(a)); a esto se le conoce como diferencia de gaussianas (DoG), con las cuales es posible determinar cuales son los puntos característicos mediante la comparación de intensidad de todos los píxeles de las DoG, con sus píxeles vecinos respectivos. Si el píxel evaluado es el valor máximo

con respecto a sus 26 vecinos, este representa un potencial punto característico. Posteriormente, se realizan pasos adicionales para garantizar la selección de puntos robustos.

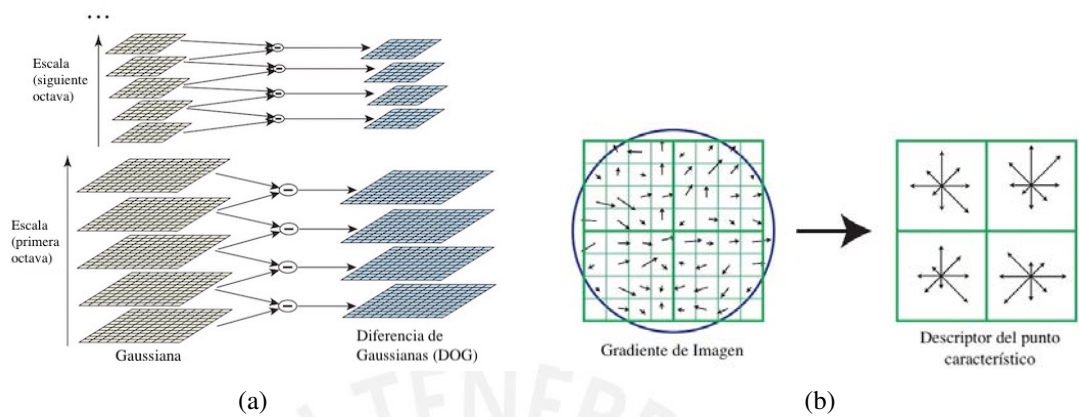


Figura 2.7: Representaciones del algoritmo SIFT, imágenes extraídas de [35]. En (a) se muestra la representación de la obtención de DoG. En (b) se observa una representación de los histogramas de una imagen, los cuales definen los descriptores SIFT.

Luego de la etapa de detección de puntos, se realiza la descripción de los puntos obtenidos. Se analizan los píxeles vecinos de cada punto característico tomando vecindarios de píxeles de 16×16 , los cuales se subdividen en bloques de 4×4 y se analizan sus orientaciones. Dentro de los cada uno de los 16 bloques formados se ordenan todas las orientaciones en histogramas y posteriormente estos valores son almacenados en un vector de 128 dimensiones. En la Figura 2.7(b) se muestra un ejemplo de este procedimiento empleando un vecindario de 8×8 píxeles.

2.4.2. Descriptores Surf:

Los descriptores Surf (*Speed up robust feature*) son descriptores que emplean la misma metodología que los descriptores SIFT pero con ciertas variaciones en su estructura. Según sus autores, estos descriptores plantean un algoritmo de detección y descripción más veloz y robusto que el algoritmo SIFT. A diferencia de los descriptores SIFT, durante la **etapa de detección** se calcula la determinante de las matrices hessianas para obtener los puntos de interés, se emplean imágenes integrales para reducir el costo computacional, asimismo, se realizan variaciones en cuanto a la descripción de los puntos característicos para que este algoritmo sea más eficiente. Mayor información puede ser encontrada en [36].

Capítulo 3

Implementación del algoritmo de estimación de pose

En los capítulos anteriores se ha planteado la problemática y se han presentado los conocimientos necesarios a tener en cuenta para el desarrollo del algoritmo propuesto. En este capítulo se explicará detalladamente el método empleado para solucionar la problemática inicial. En la primera parte se presenta el planteamiento general, tomando en cuenta el contexto y los requerimientos. Seguidamente, se describe cada sección del diagrama de bloques presentado. Finalmente se muestra una síntesis de cada una de las etapas y los recursos empleados.

3.1. Planteamiento General

3.1.1. Contexto

Este sistema de realidad aumentada tiene previsto ser implementado en la sección del recinto esquinero de la Huaca; en la Figura 3.1(a), se puede observar la sección en la que se encuentra dicho recinto. Esta parte recibe la visita de los turistas regularmente, quienes solo pueden acceder hasta cierto punto, tal como se muestra en la Figura 3.1(b). Desde este punto, los visitantes pueden capturar imágenes con un ángulo limitado, en la Figura 3.1(c) se muestra un ejemplo del tipo de fotografías que se pueden realizar. Asimismo, es necesario resaltar que esta sección posee una textura muy bien definida la cual puede servir como referencia para el sistema a implementar (ver Figura 3.1(d)).

3.1.2. Requerimientos

Según lo mencionado hasta el momento y de acuerdo al planteamiento de los objetivos, los requerimientos para la implementación del algoritmo de estimación de posición que solucione la etapa de inicialización del futuro sistema de realidad aumentada son los siguientes:

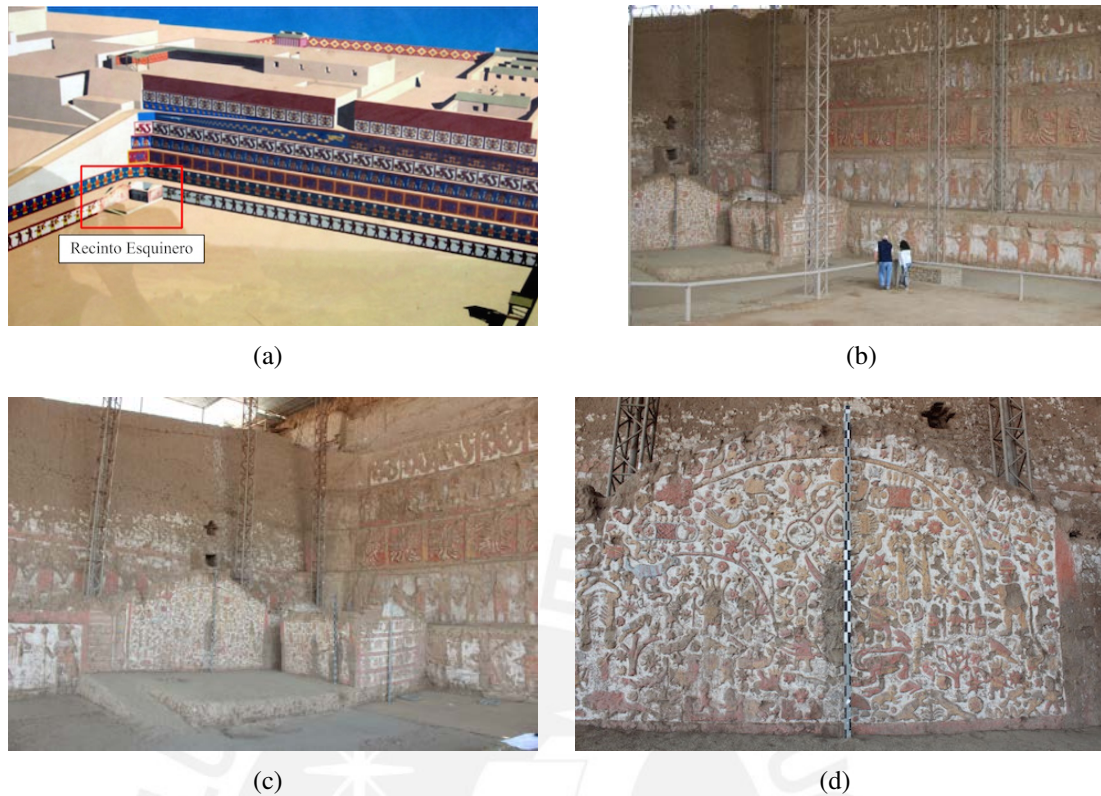


Figura 3.1: Estudio del escenario a emplear. El recinto esquinero se encuentra ubicado en la parte baja de la Huaca de la Luna (a), los turistas tienen acceso a este lugar hasta determinada posición según como se observa en (b), imagen obtenida de [39]; desde tal posición el punto de vista de un visitante es el que se muestra en la Figura (c); finalmente en (d) se muestra la textura que presenta esta sección, ideal para realizar un reconocimiento de puntos característicos.

- La exactitud de dicho algoritmo debe ser muy alta dado que se requiere la mejor experiencia del usuario. Tomando como referencia los resultados observados en estudios anteriores [40], se buscará un error de traslación relativo cercano entre 0 y 0.008 m aprox. y un error de rotación relativo entre 0 y 0.1 rad.
- El algoritmo propuesto debe ser confiable, por lo cual se buscará una tasa de estimaciones exitosas cercana al 100%.
- Finalmente, es importante que el tiempo de procesamiento sea el más rápido posible. Sin embargo, esto no se estudiará específicamente en el desarrollo propuesto.

3.1.3. Diseño

Existen diferentes soluciones al problema de estimación de pose. La elección del método escogido se basa en lo estudiado en el capítulo 2 y en el estudio del escenario a evaluar. En la Tabla 3.1 se presentan las opciones encontradas que podrían solucionar

la etapa de inicialización del sistema de RA. En esta se detallan las situaciones en las que es adecuado emplear cada método, su exactitud cualitativa y los posibles eventos ocasionadores de fallas.

Método basado en:	Adecuado cuando:	Exactitud	Posibles fallas
Marcadores (Retro-reflectivos), cámara infrarrojo - Productos Comerciales [41]	El escenario permite emplear marcadores, el costo no es un problema.	Muy alta	Muy raro
Marcadores visuales [42]	El escenario permite emplear marcadores	Preciso	Marcador no visible
Puntos de interés 2D [43]	Escenario con presencia de textura	Limitada, variable	Movimientos veloces, geometría del escenario compleja
Modelos 3D / Reconocimiento de puntos de interés 3D (Textura) [17]	Escenario 3D con textura	Limitada, variable	Movimientos veloces
Modelos 3D / Vistas-Reconocimiento de bordes [44]	Bordes notorios, escenario 3D con ausencia de textura	Preciso	Restringido a un rango de posiciones

Tabla 3.1: Análisis de métodos de estimación de pose. Métodos evaluados basados en: marcadores retro-reflectivos, marcadores visuales, puntos de interés 2D, puntos de interés 3D y bordes 3D. Comparación obtenida de [18] y [13]

De acuerdo a los requerimientos establecidos, los métodos que destacan frente a los demás son los métodos basados en modelos 3D, el método basado en puntos característicos y el método basado en la detección de los bordes.

Los métodos basados en marcadores serían una opción muy aceptable debido a la alta precisión que estos presentan y la facilidad en su implementación. Sin embargo, el escenario de la Huaca de la Luna requeriría marcadores de grandes dimensiones para que estos puedan ser detectados por las cámaras desde el punto de vista de los visitantes. Estos marcadores interferirían la apreciación del monumento arqueológico puesto que deberían ser colocados de modo permanente, de tal manera que la posición de la cámara coincida con el escenario real de manera automática.

El método basado en puntos de interés 2D podría funcionar como una alternativa al uso de marcadores, dado que se emplearían las características naturales de la Huaca de la Luna como referencia y , al igual que estos, son rápidos y fácil de implementar.

Sin embargo, emplear una imagen 2D como referencia no sería suficiente dado que los visitantes podrían desplazarse por distintos puntos y capturar imágenes diferentes a la imagen preestablecida, debido a la geometría presente en el recinto esquinero. Esto entorpecería la detección de pose y generaría errores.

Frente a estas limitaciones, los métodos basados en modelos 3D resultan una alternativa muy favorable dado que se emplean las características naturales del escenario y se considera la geometría del mismo. En esta tesis estudiará el método basado en el reconocimiento de puntos de interés 3D debido a que, ha sido empleado anteriormente en proyectos similares en donde se trabajaba con modelos 3D de gran escala. Asimismo, se observa que el método basado en vistas resulta igual de válido dado que el recinto esquinero posee bordes resaltantes y corresponde a un estudio posterior.

3.2. Etapas de la implementación del algoritmo de estimación de pose

El planteamiento general del algoritmo a implementar se encuentra ilustrado en el diagrama de bloques de la Figura 3.2. Como se puede observar, el desarrollo de este método consta de una etapa de entrenamiento (ver Algoritmo 1) y otra de detección (ver Algoritmo 2). Durante la etapa de entrenamiento se obtiene la información de la escena a detectar como la reconstrucción del modelo 3D del escenario y un conjunto de datos de puntos característicos 2D con sus correspondencias 3D. Estos puntos característicos son invariantes en escala, iluminación y en punto de visión. Por otro lado, durante la etapa de detección, se procede a detectar los puntos característicos de las imágenes de entrada y encontrar sus correspondencias empleando descriptores, con los datos obtenidos en la etapa anterior. De esta manera se obtienen correspondencias 2D-3D, con lo cual es posible emplear los métodos de estimación de pose PnP y finalmente encontrar las matrices de rotación y traslación.

A continuación se explicará detalladamente cada uno de los bloques del diseño propuesto.

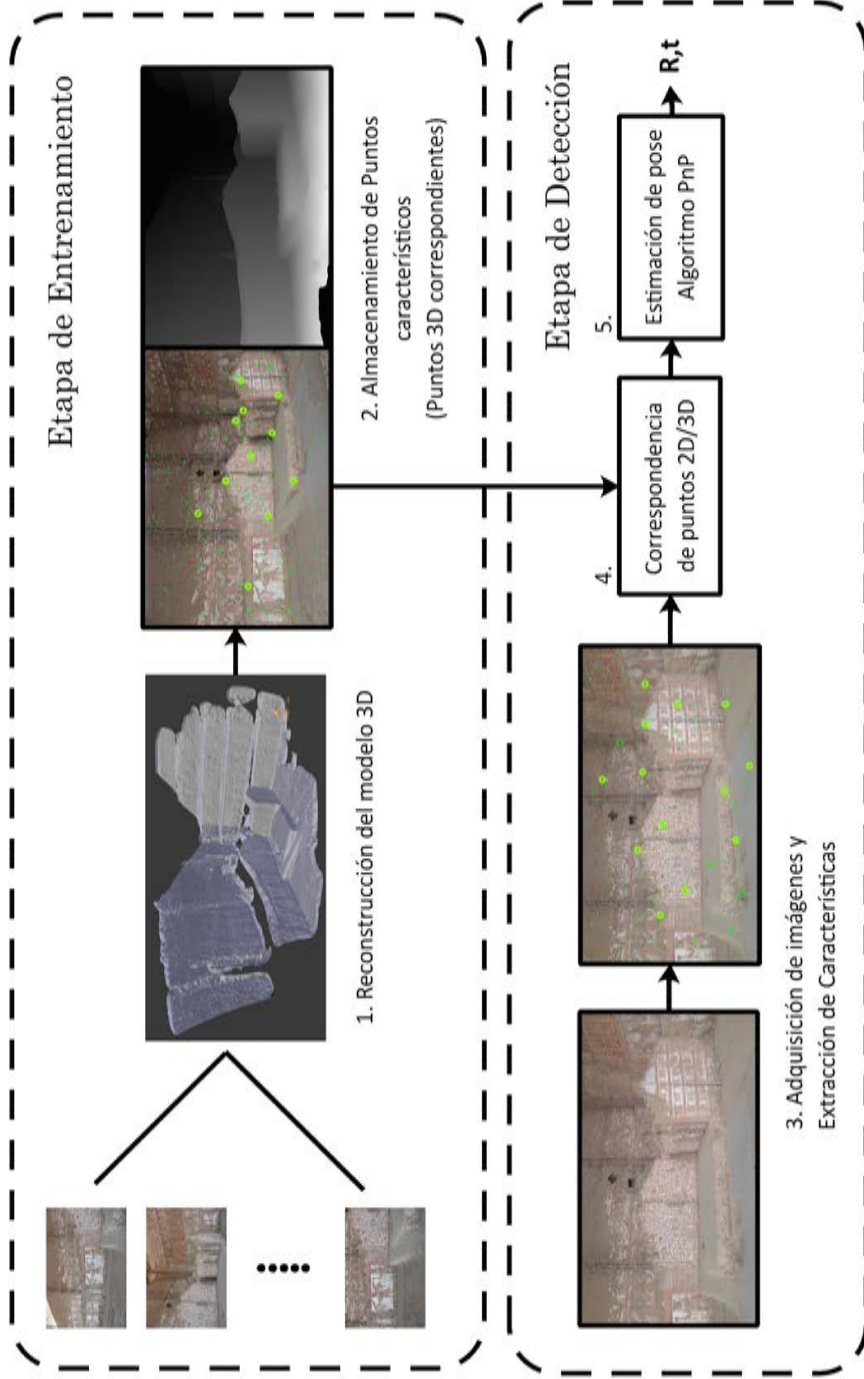


Figura 3.2: Diagrama de bloques del diseño del algoritmo propuesto. Imagen inspirada en [21]. Este diseño consta de dos etapas: una etapa de entrenamiento y una etapa de detección, la cual corresponde al proceso de inicialización del sistema de RA. Durante la primera etapa se realiza la reconstrucción del modelo 3D (1) y se almacenan los descriptores puntos característicos y sus correspondencias 3D (2). Estos descriptores son comparados con los descriptores obtenidos en la etapa de detección (3) y se consigue una correspondencia de puntos 2D con puntos 3D (4). Finalmente se emplea un algoritmo de estimación de pose PnP (5) para encontrar los parámetros de rotación y traslación.

Algoritmo 1 Etapa de entrenamiento

Entrada:Colección de imágenes: $I = \{I_1, I_2, \dots, I_n\}$ **Salida:**Colección de puntos característicos 2D: $X = \{X_1, X_2, \dots, X_m\}$,Colección de descriptores de los puntos característicos: $D = \{D_1, D_2, \dots, D_m\}$,Colección de puntos 3D proyectados: $P = \{P_1, P_2, \dots, P_m\}$

Paso 1: Creación de un modelo 3D a partir de las n imágenes de entrada (I). Se obtiene un modelo reconstruido con la posición de cada una de las cámaras de las imágenes de entrada.

Paso 2: A partir del modelo 3D, se exportan los parámetros intrínsecos $K = \{K_1, K_2, \dots, K_n\}$ y extrínsecos $RT = \{RT_1, RT_2, \dots, RT_n\}$ de cada una de las cámaras virtuales así como los mapas de profundidad de cada una de las imágenes $Z = \{Z_1, Z_2, \dots, Z_n\}$.

Paso 3: Extracción de puntos característicos 2D, descriptores y los puntos 3D correspondientes de las imágenes de entrada:

for (I_i, Z_i, K_i, RT_i) in (I, Z, K, RT) **do** $X' = \{X_1, X_2, \dots, X_k\} \leftarrow \text{DetecciónPtosCaracterísticos}(I_i)$ ▷ Ver sec. 3.2.2.a $D' = \{D_1, D_2, \dots, D_k\} \leftarrow \text{DetecciónDescriptores}(X', I_i)$ ▷ Ver sec. 3.2.2.a**for** X_j, D_j in (X', D') **do**Almacenar X_j en X ,Almacenar D_j en D , $P_j \leftarrow \text{ProyecciónPto2D_Pto3D}(X_j, Z_i, K_i, RT_i)$ ▷ Ver sec. 3.2.2.bAlmacenar P_j en P **end for****end for**

Algoritmo 2 Etapa de detección

Entrada:

Datos de entrenamiento:

- Colección de puntos característicos 2D: X ,
- Colección de descriptores de los puntos característicos: D ,
- Colección de puntos 3D proyectados: P ,

Imagen de entrada: I ,Parámetros intrínsecos de la cámara de entrada: K **Salida:**

Parámetros extrínsecos de la cámara de entrada:

- Rotación: R ,
- Traslación: T

Paso 1: Pre-procesamiento de la imagen de entrada:

▷ Ver sec. 3.2.3

$$I', K' \leftarrow \text{PreprocesamientoImg}(I, K)$$

Paso 2: Extracción de puntos característicos y descriptores de la imagen de entrada:

▷ Ver sec. 3.2.3

$$X' = \{X'_1, X'_2, \dots, X'_n\} \leftarrow \text{DetecciónPtosCaracterísticos}(I')$$

$$D' = \{D'_1, D'_2, \dots, D'_n\} \leftarrow \text{DetecciónDescriptores}(X', I')$$

Paso 3: Búsqueda de correspondencias entre los puntos característicos detectados con los puntos característicos de entrenamiento.

▷ Ver sec. 3.2.4

$$\left(\begin{array}{l} \text{TrainingDataIndexMatches}, \\ \text{DetectionDataIndexMatches}, \\ D_{[\text{TrainingDataIndexMatches}]}, \\ D'_{[\text{DetectionDataIndexMatches}]} \end{array} \right) \leftarrow \text{FindingMatches}(D, D')$$

Paso 4: Con los índices de correspondencia detectados,

▷ Ver sec. 3.2.5

tomamos los puntos característicos $X'_{[\text{DetectionDataIndexMatches}]}$ y los puntos 3D correspondientes $P_{[\text{TrainingDataIndexMatches}]}$ para obtener los parámetros extrínsecos de la imagen de entrada a partir de un algoritmo PnP.

$$(R, T) \leftarrow \text{PerspectiveNPoint} \left(\begin{array}{l} X'_{[\text{DetectionDataIndexMatches}]}, \\ P_{[\text{TrainingDataIndexMatches}]} \end{array} \right)$$

end

3.2.1. Modelamiento 3D (1)

Existen distintas técnicas empleadas en la reconstrucción de modelos 3D de patrimonio cultural, tal como se observa en [45] y [46]. Esta puede ser realizada mediante escaneo laser, o empleando sensores kinect, por ejemplo. En este caso se empleó la técnica de fotogrametría, conocida también como Structure From Motion (SFM). Para ello se utilizó el software Agisoft PhotoScan Pro [47], un programa capaz de generar contenido 3D a partir de un conjunto de imágenes. El procedimiento que se emplea para la reconstrucción dicho modelo 3D es el siguiente:

- Importación de un conjunto de imágenes en alta resolución.
- Ejecución del registro de imágenes y obtención las posiciones de la cámara.
- Generación de nube de puntos.
- Generación de malla de puntos.
- Generación de la textura del modelo 3D.
- Escalamiento del modelo 3D empleando distancias de referencias reales.

En la Figura 3.3 se muestra el modelo 3D reconstruido, así como las distintas posiciones de la cámara encontradas mediante el software. Se observa también las distancias referenciales de escalamiento. Estas fueron de 4 m. para la línea que une los puntos 1 y 3 y de 0.9 m. en el caso de la línea que une los puntos 1 y 2. Para la reconstrucción de este modelo se emplearon 36 imágenes capturadas con una cámara Canon EOS REBEL T3i.

3.2.2. Almacenamiento de puntos característicos 3D (2)

En este bloque se explica el último paso de la etapa de entrenamiento. Empleando el modelo 3D reconstruido, es posible seleccionar un conjunto de puntos 3D los cuales serán utilizados posteriormente en la comparación con los puntos característicos de las imágenes de entrada durante la etapa de detección. Para que sea posible dicha comparación, es necesario poseer los descriptores de estos puntos 3D. Debido a esto, se plantea renderizar una imagen del modelo 3D y obtener el mapa de profundidad de aquella imagen. Luego de ello, se extraerán los descriptores de los puntos característicos 2D y se obtendrán los puntos 3D realizando una proyección inversa, empleando la imagen de profundidad.

Se empleó el software libre Blender [48] como plataforma para extraer dichas imágenes del modelo 3D generado en la etapa (1). En esta plataforma se crea una cámara virtual con parámetros intrínsecos y extrínsecos conocidos, en el mismo

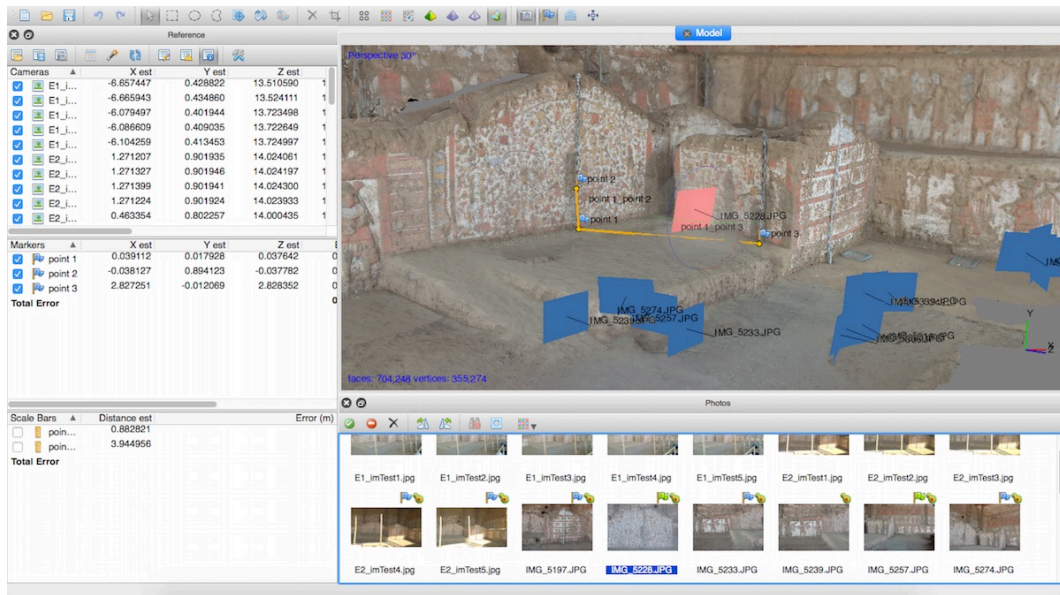


Figura 3.3: Modelamiento 3D empleando el software Agisoft PhotoScan. En la Figura se observan las imágenes registradas para la reconstrucción del modelo así como las distancias empleadas para escalar el modelo 3D, 4 m (puntos 1 y 3), 0.9 m (puntos 1 y 2).

sistema de coordenadas que el del modelo 3D. Desde esta cámara virtual se renderiza la imagen de entrenamiento.

Este bloque se subdivide en tres procedimientos. El primero de ellos es el de **extracción de características**, el segundo es el de **proyección inversa** de puntos 2D en puntos 3D y el finalmente, el **almacenamiento de los puntos obtenidos**.

a) Extracción de características

En esta etapa se pretende seleccionar puntos de control, los cuales posteriormente serán comparados con los puntos de control de las imágenes de entrada durante la etapa de detección. Se definen dos pasos fundamentales en este desarrollo. El primero es la **detección de puntos característicos** y el segundo es la **extracción de descriptores**. Este último paso se realiza para poder comparar los puntos detectados.

En el algoritmo desarrollado, se estudian dos tipos de descriptores (SIFT y SURF); debido a que son los más robustos [31], veloces y han sido empleados en proyectos similares de detección de pose y objetos como se observa en [32] y [40].

Las implementaciones de los algoritmos SIFT y SURF empleadas para este proyecto fueron obtenidas de la librería de código abierto OpenCV [49].

b) Proyección de puntos 2D a 3D

Luego de que se han obtenido los puntos deseados, se procede a obtener sus proyecciones 3D utilizando la imagen de profundidad correspondiente. Esto se logra

realizando una proyección inversa de los puntos 2D encontrados.

Para la explicación de esta operación, se tomará como ejemplo la posición de un punto (u,v) y su valor de profundidad d .

El primer paso es llevar el punto 2D al sistema de coordenadas del universo, para ello se asume que la cámara está en el origen del sistema de coordenadas y se realiza la siguiente operación:

$$X_c = d * K^{-1} * \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad (3.1)$$

donde K es la matriz de calibración de la cámara. Luego de esta proyección inversa es necesario llevar estos puntos 3D a su posición real, esto es posible empleando la matriz de rotación R y el vector de traslación t de la siguiente manera:

$$X = R^{-1} * \left(\begin{bmatrix} X_{c1} \\ X_{c2} \\ X_{c3} \end{bmatrix} - \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \right) \quad (3.2)$$

c) Almacenamiento de los puntos obtenidos

Los puntos característicos obtenidos, así como los puntos 3D correspondientes deberán ser almacenados como elementos de entrenamiento. De modo que, cuando se realice la etapa de detección, esta información pueda ser cargada previamente.

3.2.3. Adquisición y procesamiento de imágenes de entrada (3)

Este bloque representa el primer paso en la **etapa de detección**. Dado que se tiene pensado implementar el sistema de realidad aumentada como una aplicación para celulares, para la adquisición de imágenes de entrada se considerarán imágenes provenientes de smartphones las cuales pueden variar entre 2 Mpx hasta 12 Mpx de resolución.

La adquisición de imágenes de entrada implica la obtención de los **parámetros intrínsecos** y **coeficientes de distorsión** de la cámara, el **escalamiento** de las imágenes y de los parámetros intrínsecos obtenidos, la **rectificación** de las imágenes con los coeficientes de distorsión y la **extracción** de características de estas.

La calibración de la cámara se realizó empleando la herramienta de calibración de cámaras de Matlab [26], brindándole 20 imágenes de calibración. Con ello se obtuvo los parámetros intrínsecos $f_x, f_y, c_x, c_y, skew$; así como los coeficientes de distorsión $K1, K2$ y $K3$.

Debido a que las imágenes provenientes del smartphone son de alta resolución se realizará un escalamiento a las dimensiones de la imagen con el fin de disminuir el tiempo de procesamiento. Sin embargo, al momento de realizar dicho ajuste, también se debe modificar los parámetros intrínsecos. Considerando un factor de escalamiento s , donde $s > 0$. La nueva matriz de parámetros intrínsecos será la siguiente:

$$\begin{bmatrix} f_x * s & 0 & c_x * s \\ 0 & f_y * s & c_y * s \\ 0 & 0 & 1 \end{bmatrix}$$

Los coeficientes de distorsión no se ven afectados por el escalamiento y continuarán siendo los mismos.

Con las imágenes escaladas y libres de distorsión se realiza la tarea de extracción de características, la cual es idéntica a la que se realizó durante la etapa de entrenamiento. Se evaluarán los dos tipos de descriptores ya mencionados (SIFT y SURF).

3.2.4. Correspondencia de Puntos (4)

En este bloque se realiza la comparación entre los descriptores de los puntos característicos de entrenamiento con los de los puntos característicos de detección.

El método que se empleó para realizar esta etapa fue el algoritmo de búsqueda de vecinos más próximos [35] disponible en la librería de OpenCV, el cual consiste en encontrar la menor distancia entre pares de vectores correspondientes a los descriptores de la imagen de entrenamiento con los de las imágenes de entrada.

En la Figura 3.4 se puede observar un ejemplo de la correspondencia entre los puntos característicos de una imagen de entrada con la imagen de entrenamiento empleando descriptores SURF.

Esta correspondencia de puntos se realiza a través de puntos 2D; sin embargo, dado que se conoce la ubicación 3D de cada uno de los puntos de entrenamiento, es posible obtener las correspondencias entre puntos 2D con puntos 3D.

3.2.5. Estimación de pose (5)

La estimación de pose es la etapa final del proceso de detección. En esta etapa se obtienen los parámetros de rotación y de traslación a través de la solución del problema Perspective n Point, empleando las correspondencia de puntos 2D/3D obtenidas en las etapas anteriores. Según como se describió en el Capítulo 2, existen diferentes algoritmos capaces de solucionar este problema. En la presente tesis se evaluará dos de estos algoritmos: Efficient Perspective N Point (EPnP) y Robust Efficient Procrustes PnP (REPPnP). Ambos algoritmos son soluciones cerradas que presentan una mayor

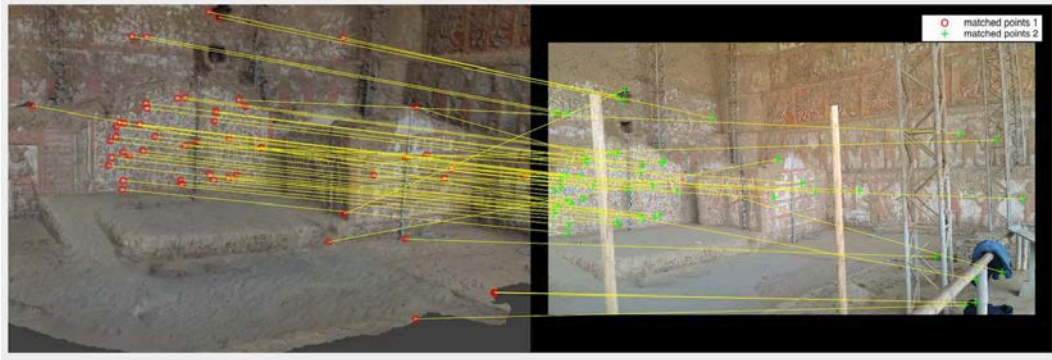


Figura 3.4: Ejemplo de correspondencia de puntos característicos empleando descriptores SURF. A la izquierda, imagen de entrenamiento. A la derecha, imagen de entrada de celular. En líneas amarillas se muestra las correspondencias obtenidas

precisión en comparación con los otros métodos [32]. Del mismo modo, el tiempo de procesamiento que requieren ambos planteamientos es mucho menor comparado con los demás. Cabe resaltar que estos métodos han sido empleados en proyectos similares anteriormente [21], [22].

Asimismo, se determinó experimentalmente el número de correspondencias óptimo para encontrar la mejor estimación.

a) Detección de outliers o anomalías

Se conoce como outliers a los valores atípicos dentro de un conjunto de datos; en otras palabras, aquellos valores numéricamente distantes del resto de los datos. En la Figura 3.4, se observan todas las correspondencias obtenidas en la etapa previa; sin embargo, no todas estas correspondencias son correctas. Estos valores incorrectos afectan el resultado cuando se realiza la estimación de pose.

El algoritmo más conocido para detectar outliers es el algoritmo RANSAC [34]. Este es un método iterativo no determinista, en el que se obtiene cierto modelo matemático a partir de una serie de iteraciones y comparaciones empleando diferentes conjuntos de datos de entrada.

En el contexto de estimación de pose, se toma un conjunto de correspondencias como los valores de entrada y se obtienen los valores de Rotación y Traslación. Estos resultados son comparados unos con otros durante un número de iteraciones definidas o cuando el error obtenido es menor a un umbral determinado. Para ello, se toman conjuntos de correspondencias diferentes en cada iteración. Para la implementación de esta etapa, se tomó en consideración los experimentos realizados por los autores de [50], por ello el número de correspondencias considerado durante cada iteración es de 7. En la Figura 3.5, se observa la detección de outliers empleando el algoritmo de estimación EPnP.

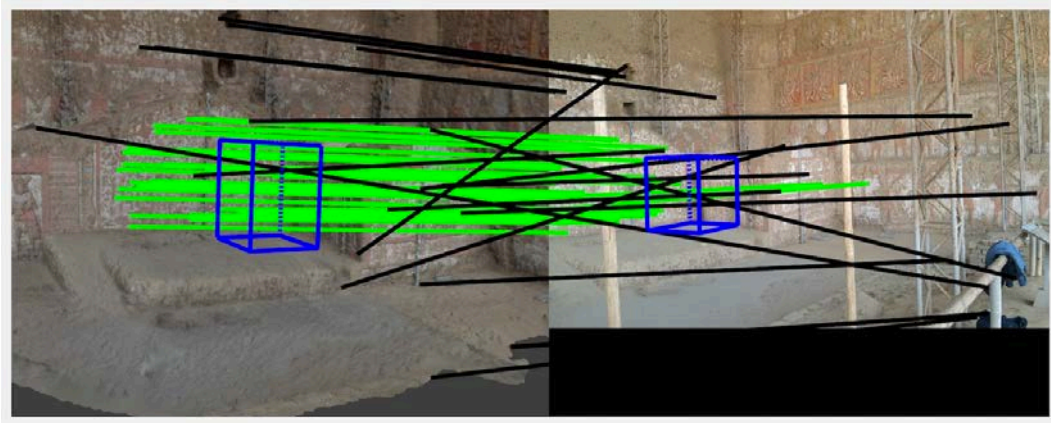


Figura 3.5: Luego de emplear el algoritmo RANSAC implementado, se obtiene una estimación de posición a partir de un conjunto de correspondencias libre de outlier. De acuerdo a los colores, las líneas verdes representan los inliers; mientras que las líneas negras, los outliers. Las líneas azules representan la proyección de un cubo, que fue posicionado en ese lugar para realizar pruebas preliminares.

3.2.6. Recursos empleados en la implementación

Para la implementación del algoritmo propuesto se emplearon recursos de distintos proyectos e investigaciones. En la Tabla 3.2 se detallan dichos recursos empleados en cada etapa del algoritmo.

Bloque	Etapas	Recursos
1. Reconstrucción del modelo 3D	Registro de imágenes y reconstrucción del modelo 3D	Agisoft PhotoScan [47]
	Obtención de imagen renderizada y mapa de profundidad a partir del modelo 3D	Blender [48]
2. Almacenamiento de puntos característicos	Extracción de características	OpenCV (SIFT) [49] OpenCV (SURF)
	Proyección de puntos 2D – 3D	Script propio
	Almacenamiento de Datos	
3. Adquisición de imágenes y Extracción de características	Adquisición de Imágenes	Smartphone LG G4, Cámara 16MPx OIS 2.0 Laser Autofocus F1.8
	Extracción de Características	OpenCV (SIFT, SURF) [49].
4. Correspondencia de Puntos	Matching (2D/3D)	OpenCV(KNN).
5. Estimación de pose - Algoritmo PnP	Algoritmo PnP	EPNP [31] + RANSAC
		REPPNP [32] - Matlab

Tabla 3.2: Descripción de los recursos empleados por cada etapa en el desarrollo del algoritmo.

Capítulo 4

Experimentos y resultados

En este capítulo se mostrarán los resultados de las pruebas realizadas para la selección de los métodos empleados en el diseño del algoritmo, específicamente, en el bloque de detección. Asimismo, se mostrarán los resultados finales del algoritmo completo evaluándose la exactitud y la precisión en los parámetros de rotación y traslación, el error de reproyección, y el tiempo de procesamiento.

4.1. Método de Validación

Basados en el esquema presentado en la sección anterior, se pueden evaluar cuatro métodos distintos. Estos presentan la misma estructura en cuanto al diseño pero cuentan con diferentes algoritmos en las etapas de **extracción de descriptores** y **estimación de pose**, (ver Figura 4.1). Estos son: empleando descriptores SIFT y método de estimación de pose EPnP (SIFT/EPnP), empleando descriptores SURF y método de estimación de pose EPnP (SURF/EPnP), empleando descriptores SIFT y método de estimación de pose REPPnP (SIFT/REPPnP) y finalmente, empleando descriptores SURF y método de estimación de pose REPPnP (SURF/REPPnP).

Se proponen dos experimentos para probar el algoritmo implementado. El primer experimento se realizó empleando imágenes sintéticas y el segundo con imágenes reales.

4.1.1. Experimento 1: Empleando imágenes sintéticas

En estas pruebas se plantea emplear un conjunto de 50 imágenes sintéticas como imágenes de prueba. Estas imágenes fueron generadas con el software Blender, de dimensiones 1328x747 píxeles, tal como se observa en la Figura 4.2. De este modo es posible utilizar los valores **ground truth** de traslación y rotación de las cámaras de prueba para poder validar los resultados.

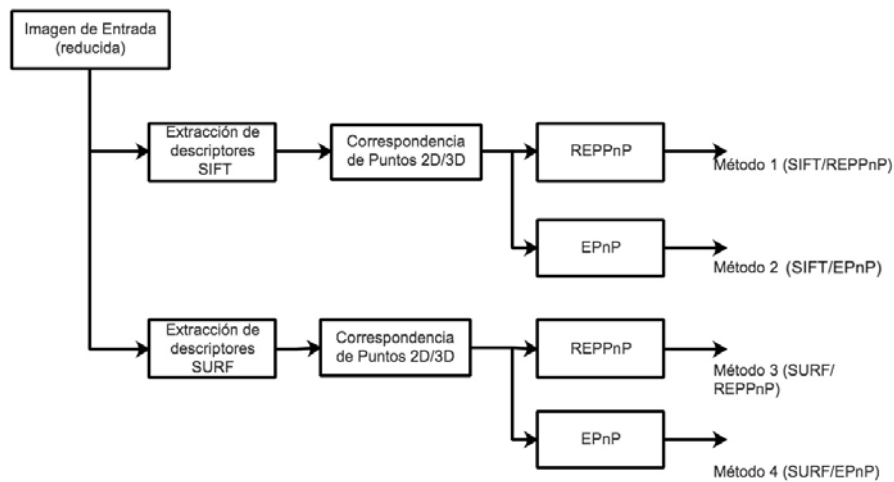


Figura 4.1: Diagrama de bloques de los métodos a evaluar SIFT/REPPnP, SIFT/EPnP, SURF/REPPnP y SURF/EPnP.

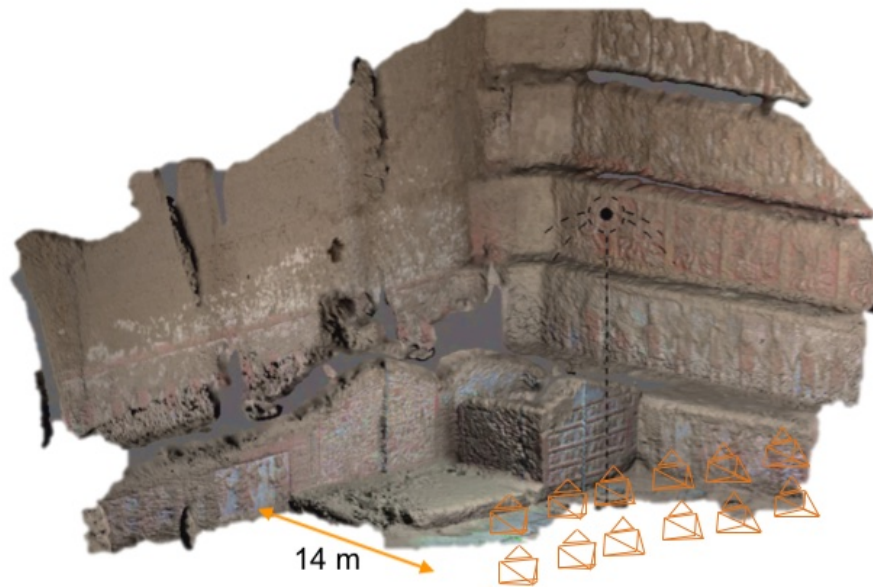


Figura 4.2: Generación de imágenes sintéticas empleando el software Blender. Se generaron 50 imágenes utilizando cámaras virtuales localizadas a 14 metros del recinto esquinero.

El objetivo de este experimento es poder comparar la robustez de los métodos evaluando los errores de **traslación**, **rotación**, la **tasa de estimaciones exitosas** y el **error de reproyección**. Asimismo, se plantea evaluar mínimo número de puntos de correspondencia 2D/3D necesario para el algoritmo. De esta manera se podrá determinar el mejor método a implementar y el umbral óptimo de puntos de correspondencia.

El **error de traslación** fue calculado mediante $e_{tras} = ||t - t_{true}|| / ||t_{true}||$, mientras que el **error de rotación** mediante $e_{rot} = ||q - q_{true}|| / ||q_{true}||$ donde q es la representación de la rotación empleando cuaterniones. El **error de reproyección** se determinó mediante el cálculo de la distancia, medida en píxeles, que existe entre las proyecciones de un mismo punto 3D empleando los valores de rotación y traslación obtenidos y de referencia. En la Figura 4.3 se observa una representación de la obtención del error de reproyección. Se definieron 11 puntos en del modelo 3D. Las líneas negras corresponden a la unión de la proyección de estos puntos empleando los parámetros Ground Truth; mientras que las líneas azules y amarillas corresponden a la unión de las proyecciones empleando los parámetros de los métodos propuestos.

Con esta prueba se consigue determinar el error en píxeles, de la estimación de pose. De esta manera es posible validar la aplicación de estos métodos en la inicialización de un sistema de RA.

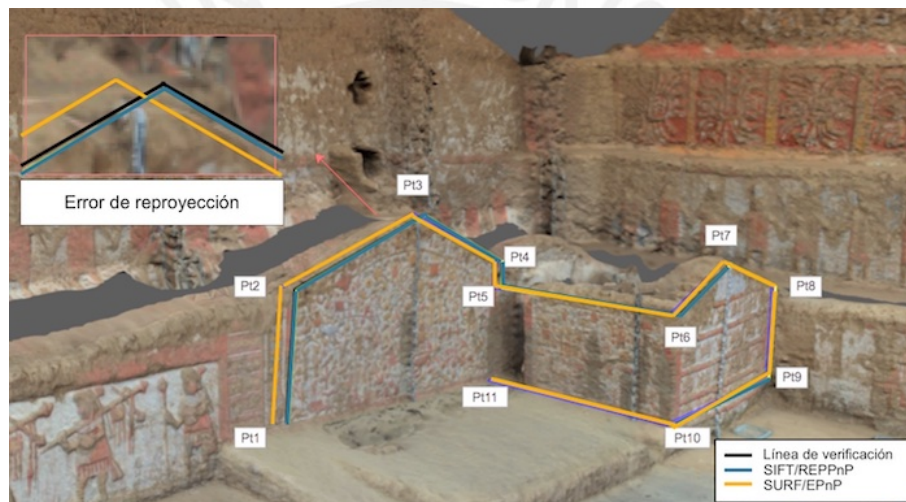


Figura 4.3: Error de Reproyección, se definen 11 puntos, los cuales servirán para determinar el error de reproyección medio de los algoritmos implementados. Las líneas negras representan los puntos de verificación; mientras que las líneas azules y amarillas representan los puntos de los algoritmos implementados.

4.1.2. Experimento 2: Empleando imágenes reales

En este experimento se plantea validar la estimación de pose empleando imágenes reales obtenidas utilizando un smartphone LG G4; asimismo, se busca evaluar los algoritmos bajo diferentes condiciones de iluminación. Durante la visita a la Huaca de la luna se pudo observar que este escenario se ve afectado por cambios en la iluminación. Se capturó un conjunto de 100 imágenes durante la mañana desde diferentes poses. En ese momento, no se presentaron sombras. Del mismo modo, durante la tarde, se capturó un conjunto de 50 imágenes; en estas condiciones si se observan sombras en la imagen. La validación de este experimento será evaluada

tomando en consideración **la tasa de estimaciones exitosas**. En cada estimación se realizará una proyección de los mismos once puntos descritos anteriormente para el error de reproyección (véase la 4.3). La unión de estos puntos simula una aplicación de RA, si las líneas están desalineadas, se considerará una estimación incorrecta.

4.2. Resultados

Todas las pruebas fueron realizadas utilizando una MacBook Pro i7, de 2,5 GHz, e implementadas en Python. El código fuente de estas pruebas se encuentra disponible en el siguiente repositorio Github [51].

4.2.1. Resultados del experimento 1

Para poder comparar los resultados obtenidos, podemos observar los gráficos de la Figura 4.4 y de la Figura 4.5. En un primer momento, se buscó determinar el número de correspondencias mínimo adecuado para el algoritmo de estimación de pose. De los resultados globales, se puede observar que con un número de correspondencia bajo, menor a 100, se obtienen resultados variables o con mucho error tanto en la traslación como en la rotación, ver la Figura 4.4 (a, b). De la misma manera, esto se ve reflejado en el error de reproyección, ver la Figura 4.4 (d).

Por otra parte, se puede visualizar que el porcentaje de estimaciones exitosas puede llegar a los picos más elevados para números de correspondencia bajos, como es el caso de los algoritmos basados en REPPnP; sin embargo este valor se mantiene y es estable para una mayor cantidad de puntos, ver la Figura 4.4 (c).

Teniendo en consideración el análisis anterior, podemos definir un **número de correspondencia mínimo de 110 puntos**, ya que este valor nos permitirá obtener errores de rotación, traslación y reproyección bajos además de una alta tasa de estimaciones exitosas.

Con este valor definido, podemos visualizar en detalle la comparación de los 4 métodos evaluados: SIFT/REPPnP, SIFT/EPnP, SURF/REPPnP y SURF/EPnP, ver la Figura 4.5.

Según los resultados obtenidos, podemos observar que los métodos basados en el algoritmo REPPnP muestran un menor error de traslación en comparación con los métodos basados en EPnP. Por otra parte, se puede ver que en cuanto al error de rotación, los algoritmos basados en EPnP son superiores a los basados en REPPnP. Para poder visualizar como afectan estos errores en nuestro sistema, podemos analizar el error máximo de reproyección, ver la Figura 4.5 (c); los resultados sugieren que los algoritmos con menor error son aquellos basados en EPnP, asimismo se puede ver que los algoritmos basados en SIFT poseen un menor error en comparación con

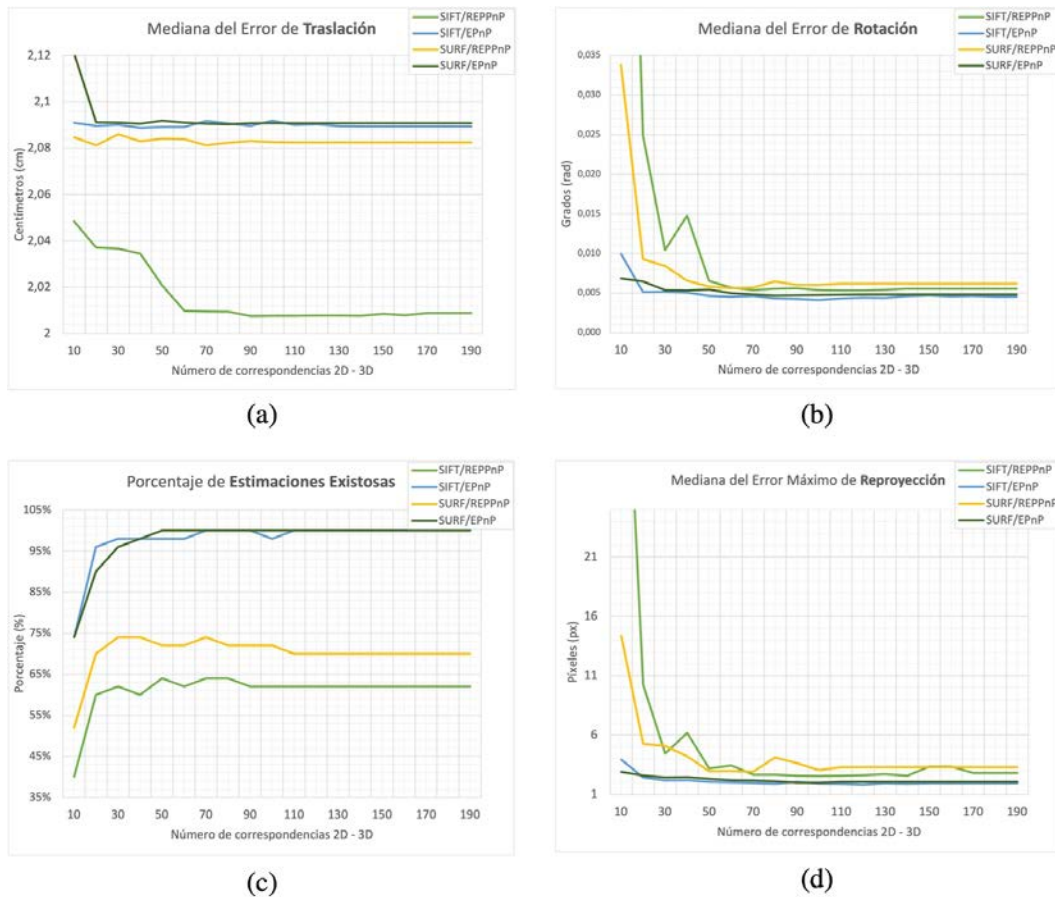


Figura 4.4: Resultados de los métodos SIFT/REPPnP, SIFT/EPnP, SURF/REPPnP y SURF/EPnP para distintos números de correspondencias. (a) Error de traslación. (b) Error de rotación. (c) Porcentaje de estimaciones exitosas. (d) Error máximo de reproyección.

aquellos basados en SURF. Por último, los resultados de la estimaciones exitosas muestran que los algoritmos basados en EPnP llegan a tener un 100% de estimaciones exitosas, a diferencia de los métodos basados en REPPnP que solo llegan hasta un 70%. Los tiempos de procesamiento promedio son mostrados en la Figura 4.5.(e) como referencia e indican que estos métodos poseen un tiempo de ejecución menor a 1 segundo.

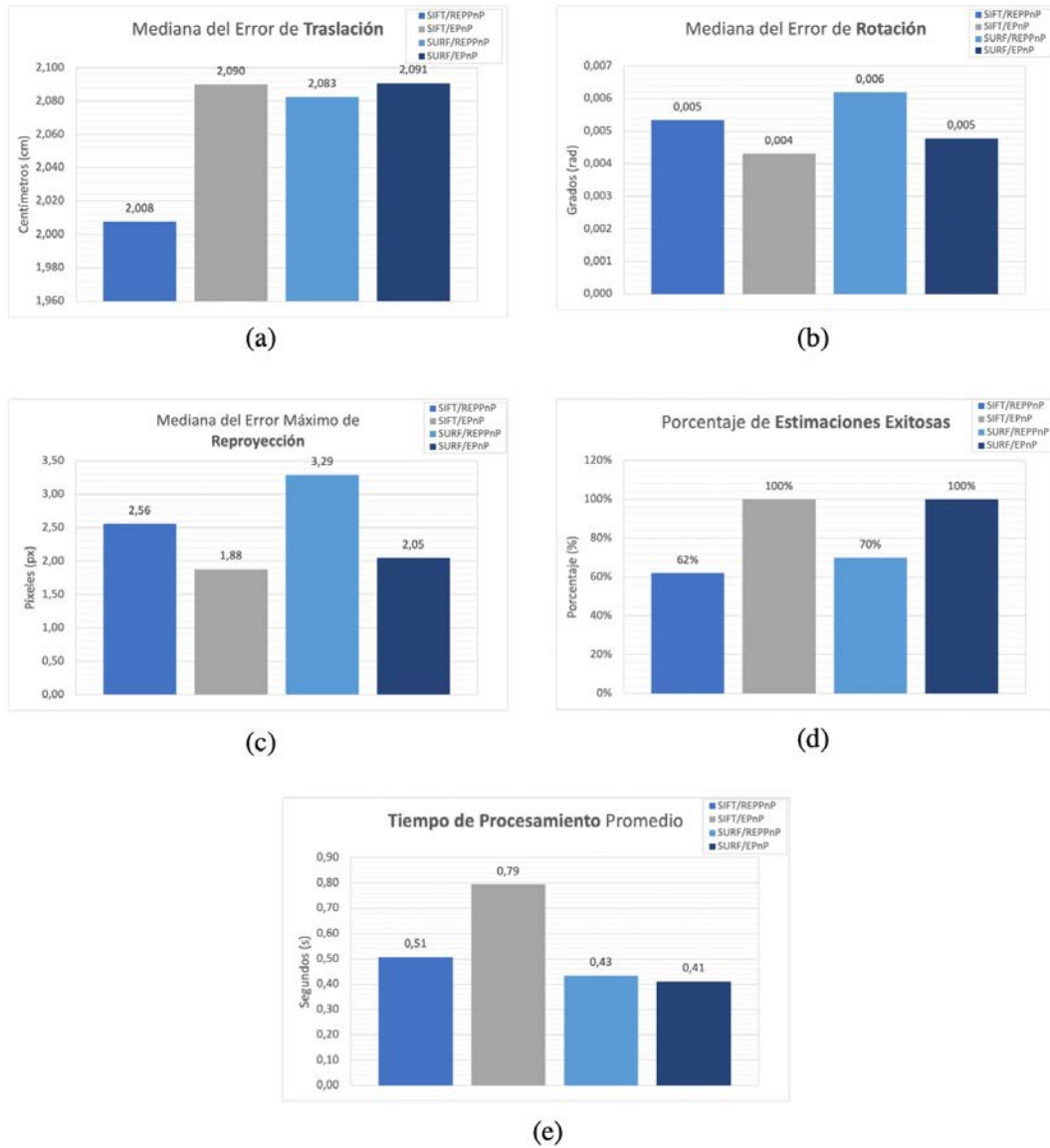


Figura 4.5: Resultados de los métodos SIFT/REPPnP, SIFT/EPnP, SURF/REPPnP y SURF/EPnP empleando 110 puntos de correspondencia 2D - 3D. (a) Error de traslación. (b) Error de rotación. (c) Porcentaje de estimaciones exitosas. (d) Error máximo de reproyección.

4.2.2. Resultados del experimento 2

En las Figuras 4.6 y 4.7 se puede observar los resultados de este experimento. Para evaluar el porcentaje de estimaciones exitosas se realizó la proyección de un objeto 3D sobre las imágenes de prueba. De esta manera, se pudo validar visualmente si una estimación era correcta o incorrecta. En las Figuras 4.7 (a, c, e) se observan las estimaciones para los casos en condiciones de iluminación regular, por otra parte en las Figuras 4.7 (b, d, f) se muestran los casos en condiciones de alta iluminación.

Los resultados obtenidos se muestran en la Figura 4.7. Según estos resultados, observamos que las implementaciones más resaltantes son aquellas basadas en EPnP ya que poseen una mayor tasa de aciertos en la estimación. Podemos observar que en condiciones regulares de iluminación estos algoritmos llegan a un 100% de aciertos, en comparación con los métodos basados en REPPnP, que llegan a un máximo de 85%. Por otra parte, se puede ver una diferencia cuando las condiciones de iluminación son más difíciles: Los algoritmos basados en EPnP siguen siendo mejores que los de REPPnP; sin embargo se observa una superioridad en los descriptores SURF con un 94% de aciertos comparados con los descriptores SIFT con un 85% de aciertos.

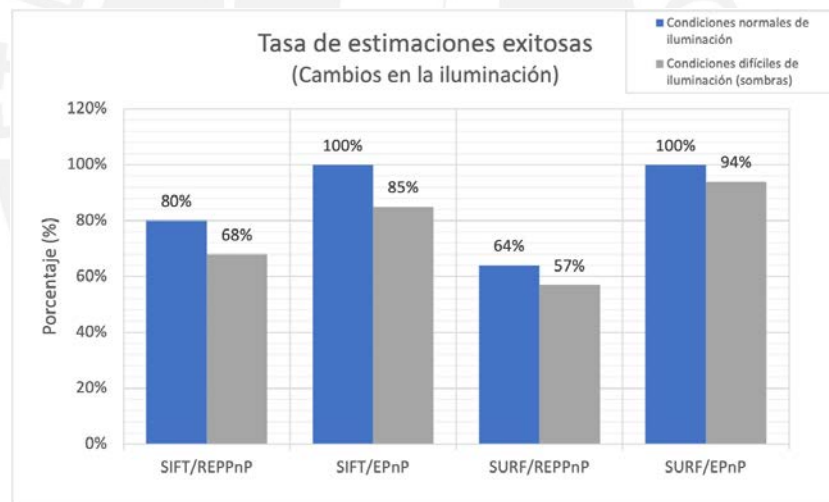


Figura 4.6: Resultados de la tasa de estimaciones exitosas para los métodos SIFT/REPPnP, SIFT/EPnP, SURF/REPPnP y SURF/REPPnP frente a imágenes con cambios de luminosidad

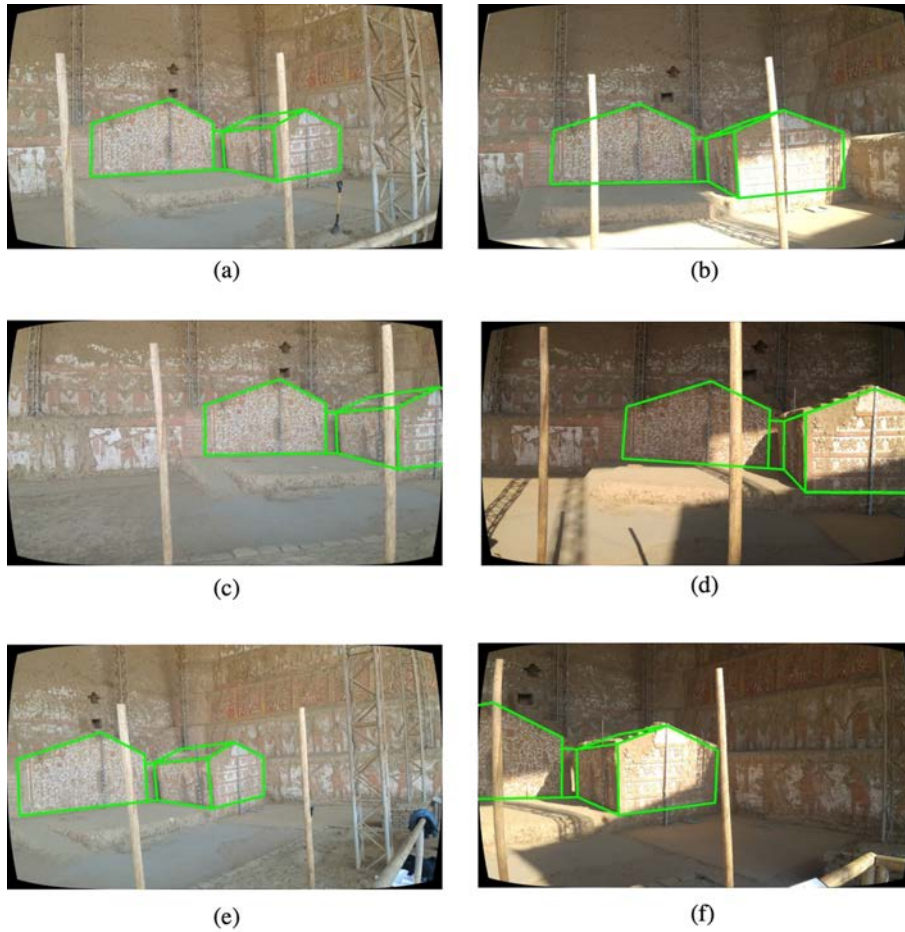


Figura 4.7: Pruebas de confiabilidad. Las imágenes (a),(c) y (e) son una muestra de las fotografías que se capturan durante la mañana; por otra parte, las imágenes (b), (d) y (f) fotografías capturadas por la tarde. Se determinó visualmente la cantidad de aciertos de la estimación de pose.

4.2.3. Discusión de resultados

Según se observó en los resultados anteriores, los métodos SIFT/EPnP y SURF/EPnP presentan los mejores resultados. Se pudo ver una ligera superioridad en cuanto a la robustez de las pruebas con imágenes reales del algoritmo SURF/EPnP frente al SIFT/EPnP. Asimismo, dado que se conoce que los descriptores SURF poseen un menor tiempo de procesamiento que los descriptores SIFT. El método seleccionado como mejor método en esta implementación, será el método SURF/EPnP.

Este método presentó un error de traslación de 2.091 cm así como un error de rotación de 0.005 rad. Asimismo, presenta un error de reproyección máximo de 2.05 píxeles y una tasa de estimaciones exitosas de 100% en condiciones normales y de 94% en condiciones de iluminación difíciles. Finalmente, este algoritmo implementado posee un tiempo de procesamiento promedio de 0.43 segundos. De esta manera, podemos validar que esta configuración es adecuada para la etapa de inicialización de un sistema de RA.

En la Figura 4.1, se presenta un cuadro comparativo con todas las pruebas realizadas. Las otras opciones de implementación como SIFT/REPPnP y SURF/REPPnP son opciones precisas y robustas ante outliers; sin embargo, el nivel de confiabilidad de estas no sería el adecuado para este proyecto.

Criterio	SIFT/REPPnP	SIFT/EPnP	SURF/REPPnP	SUR/EPnP
Error de traslación	Muy bajo	Bajo	Bajo	Bajo
Error de rotación	Bajo	Muy bajo	Bajo	Muy bajo
Error de reproyección	Bajo	Muy bajo	Bajo	Muy bajo
Tasa de estimaciones exitosas	Baja	Muy alta	Alta	Muy alta
Robustez de imágenes reales	Baja	Muy alta	Baja	Muy alta

Tabla 4.1: Análisis cualitativo de las pruebas realizadas. Los métodos SIFT/EPnP y SURF/EPnP destacan frente a los demás

Conclusiones

- Según el estudio realizado, se ha demostrado que las técnicas que mejor se adaptan para la inicialización de un sistema de RA orientado a patrimonio cultural son los métodos basados en modelos 3D; debido a que estos presentan una geometría compleja. Entre estos métodos se encuentran aquellos que están basados en bordes y los que están basados en textura.
- Se ha demostrado que el algoritmo implementado permite realizar la inicialización de un sistema de realidad aumentada, empleando solo una imagen de referencia durante la etapa de entrenamiento y un modelo 3D. Esta inicialización permitirá estimar la posición de la cámara, la cual será empleada para posicionar los objetos virtuales en la escena real de manera precisa.
- El algoritmo implementado presenta un error de traslación de 2.091 cm así como un error de rotación de 0.005 rad. Asimismo, presenta un error de reproyección máximo de 2.05 píxeles y una tasa de estimaciones exitosas de 100% en condiciones de iluminación normal y de 94% en condiciones de iluminación difíciles. Asimismo, el tiempo de procesamiento que presenta dicho algoritmo es de 0.43 segundos.
- En las pruebas realizadas se emplearon descriptores SIFT y SURF para la extracción de características, se pudo observar que ambos descriptores ofrecen una alta precisión. Por otro lado, se emplearon los algoritmos REPPnP y EPnP para la estimación de pose. Se pudo observar que el algoritmo EPnP ofrece una mayor confiabilidad comparado con REPPnP.
- Este algoritmo de inicialización puede ser empleado en un sistema de realidad aumentada complementado con algoritmos de seguimiento empleando sensores adicionales como GPS y acelerómetros. El algoritmo implementado será empleado periódicamente para detectar la posición de la cámara cuando el algoritmo de seguimiento falle y se deba recalibrar.

Recomendaciones

- En las pruebas realizadas, se ha empleado solo una imagen de referencia en la etapa de entrenamiento, capturada en la parte frontal del recinto esquinero. Es posible aumentar el número de imágenes en la etapa de entrenamiento capturadas en diferentes secciones de la Huaca de la Luna. Sin embargo, al emplear mayor cantidad de descriptores se requerirá una mayor eficiencia en la etapa de detección de correspondencias; debido a ello, se recomienda utilizar algún algoritmo como Bag of Words que pueda agilizar el proceso.
- El modelo 3D que se empleó fue realizado mediante la técnica de fotogrametría. Otra de las alternativas para reconstruir modelos 3D es el uso de un escáner 3D, con el cuál se puede obtener una mayor precisión.
- Como trabajo futuro, se debe considerar el uso de algoritmos basados en GPU para la extracción de características, con el fin de reducir el tiempo de procesamiento. Asimismo, se propone el estudio de descriptores de menor complejidad como serían los descriptores FAST o ORB.
- Como ya se mencionó en las conclusiones, se plantea emplear sensores adicionales para la etapa de seguimiento del sistema RA, este seguimiento puede ser realizado empleando algoritmos predictivos como filtro Kalman, para brindar robustez al sistema.

Bibliografía

- [1] Ronald T Azuma. A survey of augmented reality. *Presence: Teleoperators and virtual environments*, 6(4):355–385, 1997.
- [2] Zakiah Noh, Mohd Shahrizal Sunar, and Zhigeng Pan. A review on augmented reality for virtual heritage system. In *International Conference on Technologies for E-Learning and Digital Entertainment*, pages 50–61. Springer, 2009.
- [3] Eva Savina Malinverni, Francesca Colosi, and Roberto Orazi. Making visible the invisible. augmented reality visualization for 3d reconstructions of archaeological sites. In *Augmented and Virtual Reality: Second International Conference, AVR 2015, Lecce, Italy, August 31-September 3, 2015, Proceedings*, volume 9254, page 25. Springer, 2015.
- [4] Paula. Wikitude 3d tracking (beta version). <http://www.wikitude.com/blog-wikitude-3d-tracking-beta/>, 2015. Visitado el 02 de Setiembre del 2016.
- [5] Areti Damala, Isabelle Marchal, and Pascal Houlier. Merging augmented reality based features in mobile multimedia museum guides. In *Anticipating the Future of the Cultural Past, CIPA Conference 2007, 1-6 October 2007*,, pages 259–264, 2007.
- [6] M Canciani, E Conigliaro, M Del Grasso, P Papalini, and M Saccone. 3d survey and augmented reality for cultural heritage. the case study of aurelian wall at castra praetoria in rome. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 931–937, 2016.
- [7] Byung-Kuk Seo, Kangsoo Kim, Jungsik Park, and Jong-Il Park. A tracking framework for augmented reality tours on cultural heritage sites. In *Proceedings of the 9th ACM SIGGRAPH Conference on Virtual-Reality Continuum and its Applications in Industry*, pages 169–174. ACM, 2010.
- [8] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier,

- and Blair MacIntyre. Recent advances in augmented reality. *IEEE computer graphics and applications*, 21(6):34–47, 2001.
- [9] ARmedia. 3d tracker. <http://www.armedia.it/tracker.php>, 2013. Visitado el 02 de Setiembre del 2016.
- [10] Metaio. Product support. http://www.metaio.eu/product_support.html, 2015. Visitado el 15 de Noviembre del 2016.
- [11] augmentedorg. metaio’s insidear 2011. <https://www.flickr.com/photos/augmentedorg/6191846132/>, 2011. Visitado el 15 de Noviembre del 2016.
- [12] Veronica Teichrieb, Joao Paulo Silva do Monte Lima, Eduardo Lourenço Apolinário, Thiago Souto Maior Cordeiro de Farias, Márcio Augusto Silva Bueno, Judith Kelner, and Ismael HF Santos. A survey of online monocular markerless augmented reality. *International Journal of Modeling and Simulation for the Petroleum Industry*, 1(1), 2007.
- [13] João Paulo Lima, Francisco Simões, Lucas Figueiredo, and Judith Kelner. Model based markerless 3d tracking applied to augmented reality. *Journal on 3D Interactive Systems*, 1, 2010.
- [14] Harald Wuest, Florent Vial, and D Strieker. Adaptive line tracking with multiple hypotheses for augmented reality. In *Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR’05)*, pages 62–69. IEEE, 2005.
- [15] Sumit Basu, Irfan Essa, and Alex Pentland. Motion regularization for model-based head tracking. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 3, pages 611–616. IEEE, 1996.
- [16] Frédéric Jurie and Michel Dhome. A simple and efficient template matching algorithm. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 544–549. IEEE, 2001.
- [17] Luca Vacchetti, Vincent Lepetit, and Pascal Fua. Stable real-time 3d tracking using online and offline information. *IEEE transactions on pattern analysis and machine intelligence*, 26(10):1385–1391, 2004.
- [18] Vincent Lepetit and Pascal Fua. *Monocular model-based 3D tracking of rigid objects*. Now Publishers Inc, 2005.
- [19] Christian Wiedemann, Markus Ulrich, and Carsten Steger. Recognition and tracking of 3d objects. In *Joint Pattern Recognition Symposium*, pages 132–141. Springer, 2008.

- [20] Iryna Skrypnyk and David G Lowe. Scene modelling, recognition and tracking with invariant image features. In *Mixed and Augmented Reality, 2004. ISMAR 2004. Third IEEE and ACM International Symposium on*, pages 110–119. IEEE, 2004.
- [21] A Rubio, Michael Villamizar, Luis Ferraz, Adrián Peñate-Sánchez, Alberto Sanfeliu, and Francesc Moreno-Noguer. Estimación monocular y eficiente de la pose usando modelos 3d complejos. 2014.
- [22] Edgar Riba Pi. Implementation of a 3d pose estimation algorithm. 2015.
- [23] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [24] Alejandro J Troccoli. *New methods and tools for 3D-modeling using range and intensity images*. PhD thesis, COLUMBIA UNIVERSITY, 2006.
- [25] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.
- [26] Jean-Yves Bouguet. Camera calibration toolbox for matlab. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.
- [27] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003.
- [28] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2969–2976. IEEE, 2011.
- [29] Man Lee Liu and Kin Hong Wong. Pose estimation using four corresponding points. *Pattern Recognition Letters*, 20(1):69–74, 1999.
- [30] Adrian Penate-Sanchez, Juan Andrade-Cetto, and Francesc Moreno-Noguer. Exhaustive linearization for robust camera pose and focal length estimation. *IEEE transactions on pattern analysis and machine intelligence*, 35(10):2387–2400, 2013.
- [31] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155–166, 2009.

- [32] Luis Ferraz, Xavier Binefa, and Francesc Moreno-Noguer. Very fast solution to the pnp problem with algebraic outlier rejection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 501–508, 2014.
- [33] Steffen Urban, Jens Leitloff, and Stefan Hinz. Mlpnp - A real-time maximum likelihood solution to the perspective-n-point problem. *CoRR*, abs/1607.08112, 2016.
- [34] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [35] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [36] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.
- [37] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, 2014.
- [38] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. IEEE, 2011.
- [39] Sitio Oficial Huacas del Sol y de la Luna. Boletines informativos. <http://www.huacasdemoche.pe/index.php?menuid=5&submenuid=38&articuloid=87&subarticuloid=>, 2011. [Online; Fecha de consulta: 10 de noviembre del 2016].
- [40] A Rubio, Michael Villamizar, Luis Ferraz, Adrián Penate-Sanchez, Arnau Ramisa, Edgar Simo-Serra, Alberto Sanfeliu, and Francesc Moreno-Noguer. Efficient monocular pose estimation for complex 3d models. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1397–1402. IEEE, 2015.
- [41] Sitio oficial de vicon. <https://www.vicon.com/>. Accessed: 2017-08-20.
- [42] Sitio oficial de artoolkit. <http://www.hitl.washington.edu/artoolkit/>. Accessed: 2017-08-20.

- [43] Gilles Simon, Andrew W Fitzgibbon, and Andrew Zisserman. Markerless tracking using planar structures in the scene. In *Proceedings IEEE and ACM international symposium on augmented reality (ISAR 2000)*, pages 120–128. IEEE, 2000.
- [44] Joao Paulo Silva do Monte Lima, Francisco Paulo Magalhaes Simoes, Lucas Silva Figueiredo, and Judith Kelner. Model based markerless 3d tracking applied to augmented reality. *Journal on Interactive Systems*, 1(1), 2010.
- [45] Heiko Herrmann and Emiliano Pastorelli. Virtual reality visualization for photogrammetric 3d reconstructions of cultural heritage. In *International Conference on Augmented and Virtual Reality*, pages 283–295. Springer, 2014.
- [46] Matias Quintana. Registro de una secuencia temporal de nubes de puntos utilizando tecnología kinect para la reconstrucción tridimensional de material arqueológico. Junio 2014.
- [47] Agisoft PhotoScan Professional Edition. *version 1.1.0 build 2004 (64 bits)*. Agisoft LLC, 2014.
- [48] Blender.org. Software libre blender. <https://www.blender.org>, 2016.
- [49] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [50] Francesc Moreno-Noguer, Vincent Lepetit, and Pascal Fua. Accurate non-iterative o (n) solution to the pnp problem. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.
- [51] Ricardo Moisés Rodríguez Oceda (moisesr4). 3d pose estimation heritage. <https://github.com/moisesr4/3D-pose-estimation-heritage>, 2022.