



ESCUELA DE INGENIERÍAS
INDUSTRIALES



UNIVERSIDAD DE VALLADOLID
ESCUELA DE INGENIERÍAS INDUSTRIALES

TRABAJO FIN DE MASTER

Modelado automático 3D de
edificios históricos

Autor:

ROMÁN CEMBRANOS, Javier

Director:

Dr. GÓMEZ GARCÍA BERMEJO, Jaime

DEPARTAMENTO DE INGENIERÍA DE
SISTEMAS Y AUTOMÁTICA.

Valladolid, 9 de junio de 2019

Palabras clave

BIM, Nubes de Puntos, Reconstrucción 3D, Digitalización 3D, Tratamiento automático.

"La autoeducación es, estoy convencido, el único tipo de educación que existe."

Isaac Asimov

UNIVERSIDAD DE VALLADOLID

Proyecto Fin de Máster

Escuela de Ingenierías Industriales

Departamento de Ingeniería de Sistemas y Automática.

Investigación en Ingeniería de Procesos y Sistemas Industriales

Modelado automático 3D de edificios históricos

por ROMÁN CEMBRANOS, Javier

Resumen

En el presente Trabajo Fin de Máster se aborda el desarrollo de una metodología que, utilizando algoritmos basados en técnicas de visión artificial, permita generar la reconstrucción virtual de una edificación de forma automática a partir de una nube de puntos. La metodología parametriza los elementos básicos del modelo virtual sin que sea necesaria la intervención de un experto en ningún momento.

La metodología propuesta contempla varias fases de procesamiento. Para cada una de estas fases se han desarrollado algoritmos que son capaces de detectar y modelar de forma automática los elementos arquitectónicos más representativos de edificaciones históricas. Concretamente, los algoritmos han sido optimizados para el análisis de los muros y las torres de edificaciones defensivas como los castillos, detectados aplicando técnicas de segmentación y etiquetado inteligente.

Los diferentes algoritmos de la metodología propuesta han sido integrados en una herramienta informática. Los algoritmos, que forman parte de una biblioteca software, pueden ser utilizados directamente en el entorno BIM Revit de la compañía Autodesk, lo que mejora notablemente la usabilidad del desarrollo realizado.

Abstract

In the present Master's Thesis, the development of a methodology that, using algorithms based on artificial vision techniques, allows generating the virtual reconstruction of a building automatically from a cloud of points is presented. The methodology parameterizes the basic elements of the virtual model without requiring the intervention of an expert at any time.

The proposed methodology contemplates several processing phases. For each of these phases, several algorithms have been developed that are able to automatically detect and model the most representative architectural elements of heritage buildings. Specifically, the algorithms have been optimized for the analysis of walls and towers of defensive buildings such as castles, detected by applying segmentation and intelligent labeling techniques.

The algorithms of the proposed methodology have been integrated into a computer tool. The algorithms, which are part of a software library, can be used directly within the BIM environment Revit property of the Autodesk company, which significantly improves the usability of the proposed methodology.

Agradecimientos

Este proyecto no se podría haber llevado a cabo sin la colaboración de las personas que me han brindado su tiempo, ayuda y conocimientos y a las cuales quiero dedicar las siguientes líneas.

Quedo especialmente agradecido con el Doctor Pedro Martín Leronés, quien me ha brindado sabios consejos siempre que los he necesitado y sin los cuales la idea de desarrollar este proyecto habría permanecido durmiendo en el fondo de un oscuro cajón.

A mi compañero Álvaro Melero le agradezco la ayuda prestada en esos numerosos momentos en los que el código no quería compilar y mucho menos funcionar y la paciencia que ha tenido conmigo cada una de las veces que he aparecido por su mesa con un nuevo problema.

Finalmente quiero agradecerle a mi compañero José Llamas que confiara en mí para desarrollar una aplicación de tratamiento de nubes de puntos 3D, campo inexplorado para mí en aquel entonces, cuando empecé a trabajar con él. Aquel encargo plantó la semilla de lo que tres años más tarde daría lugar a la publicación de mi primer artículo y la realización de un trabajo de final de máster.

Y como no podía ser de otra forma, agradezco la participación activa de mi tutor, Jaime Gómez García-Bermejo. Agradezco sus respuestas, directrices, comentarios y la implicación mostrada en este proyecto. Me ha asesorado siempre que lo he necesitado.

*Dedicado a mis padres que me han apoyado y ayudado y sin los que
no podría haber llegado hasta donde me encuentro hoy.*

Índice general

Resumen	IV
Agradecimientos	VI
Contenidos	IX
Índice de Figuras	XI
Índice de Tablas	XIII
Abreviaturas	XV
1. Introducción	1
1.1. Introducción	3
1.2. Objetivo	3
1.3. Estado de la tecnología	4
1.4. Caso de estudio	7
1.5. Antecedentes	9
1.6. Estructura del documento	9
2. Tecnologías de digitalización tridimensional	11
2.1. Clasificación de las técnicas de digitalización	13
2.2. Representación de modelos 3D	15
2.2.1. Nube de puntos	15
2.2.2. Malla de triángulos	17
2.3. Digitalización aplicada a patrimonio cultural	18
2.3.1. Escaneado 3D	18
2.3.2. Fotomodelado	19
2.3.3. Comparación de tecnologías	21
2.3.3.1. Precisión y resolución	21
2.3.3.2. Rango de medida	22
2.3.3.3. Tiempos	22
2.3.3.4. Precio	23
2.4. El castillo de los Comuneros de Torrelobatón	24

3. Entorno de trabajo BIM	27
3.1. BIM en proyectos de patrimonio	29
3.2. Herramientas comerciales para el modelado BIM de patrimonio	31
3.2.1. Importación de nubes en Revit	31
3.2.2. Importación de nubes en ArchiCAD	32
3.3. Metodología de modelado con herramientas comerciales	35
4. Metodología de modelado automático	41
4.1. Tratamiento de nubes de puntos	43
4.1.1. Herramientas comerciales	43
4.1.2. Biblioteca PCL	44
4.2. Algoritmos para la interpretación de nubes de puntos	45
4.2.1. Importación	46
4.2.2. Preprocesado	47
4.2.3. Segmentación	48
4.2.4. Clasificación	51
4.2.5. Parametrización	54
4.2.5.1. Muros	54
4.2.5.2. Torres	58
4.3. Integración con Revit	60
4.3.1. Interfaz	60
5. Resultados	65
5.1. Ensayos realizados	67
5.2. Nube de puntos sintética	69
5.3. Nube de puntos real	72
5.4. Modelo BIM generado	74
6. Estudio económico	77
6.1. Costes directos	79
6.1.1. Coste de personal	79
6.1.2. Coste de amortización de equipos	80
6.1.3. Coste de amortización de software	81
6.1.4. Coste de material fungible	81
6.1.5. Costes directos totales	82
6.2. Coste indirecto	82
6.3. Coste total	82
7. Conclusiones y líneas futuras	85
7.1. Conclusiones	87
7.2. Líneas futuras	88
7.2.1. Ampliación del clasificador	88
7.2.2. Detección de aperturas	88
Bibliografía	91

Índice de figuras

1.1. Wang y otros. Caso de estudio. a) Nube original b) Nube segmentada. c) Detección de límites y bordes d) Ajuste de geometría e) Modelo semántico final	5
1.2. Zolanvari y Laefer. Caso de estudio. a) Nube original b) Segmentación horizontal c) Segmentación vertical d) Bordes detectados e) Dibujo con mediciones	5
1.3. Li y otros. Caso de estudio. a) Nube de puntos b) Segmentación c) Reconstrucción 3D d) Modelo final	6
1.4. Stambler. Caso de estudio. a) Nube de puntos. b) Segmentación. c) Superficies coloreadas por clases. d) Volumen final con nube de puntos de intensidades. e) Plano en planta.	6
1.5. Ochmann. Caso de estudio.	7
1.6. Digitalización del castillo de Torrelobatón	8
2.1. Clasificación de las tecnologías de digitalización más representativas. . . .	15
3.1. Nube de puntos importada en Revit	32
3.2. Vista en planta de la nube de puntos del Hospital de los Inocentes, Florencia (Italia), en ArchiCAD. (Fuente: Leronés y otros [18]).	34
3.3. Diagrama de bloques del proceso seguido para realizar un modelo 3D y a partir de una nube de puntos. Naranja: primer paso; amarillo: segundo paso; Verde: tercer paso; Azul: resultados. (Fuente: López y otros [20]). . . .	36
3.4. Nube de puntos de la Iglesia de San Nicolás, Obergum (Países Bajos). (Fuente: Leronés y otros [18]).	38
3.5. Modelo BIM de la Iglesia de San Nicolás en ArchiCAD y exportado a formato IFC2x3. (Fuente: Leronés y otros [18]).	39
4.1. Estructura de la biblioteca PCL. (Fuente: web oficial de la biblioteca PCL[29]).	45
4.2. Esquema general del método propuesto. De izquierda a derecha. Preprocesado; Segmentación; Clasificación; Parametrización.	46
4.3. Diagrama de flujo de la metodología propuesta para el cálculo de los vértices de los muros.	58
4.4. Cinta de selección de plug-ins de Revit (parte superior izquierda).	61
4.5. Pantallas de configuración del plug-in.	61
4.6. Formulario de selección de nube de puntos	62
5.1. Nube sintética. a) Preprocesado; b) Segmentado; c) Clasificado; d) Modelo final.	69

5.2. Nube del castillo. a) Preprocesado; b) Segmentado; c) Clasificado; d) Puntos del modelo y puntos no ajustados en gris; e) Puntos proyectados; f) Parametrización final y bordes mejorados (detalle)	73
5.3. a) Segmentado; b) No segmentado; c) Proyección de puntos extraídos de revolución; d) Perfil poligonal.	74
5.4. Modelado BIM en Revit. a) Nube de puntos de partida; b) Modelo final generado automáticamente.	75

Índice de tablas

2.1. Características de la nube de puntos del Castillo de Torrelobatón	24
4.1. Tipos de puntos soportados nativamente por la biblioteca PCL	46
4.2. Vértices teóricos	56
4.3. Ejemplo de valores típicos de distancias	57
4.4. Conjunto típico de vértices para un muro rectangular	57
4.5. Parámetros de configuración del plug-in para Revit	62
5.1. Selección de parámetros y tiempo empleado en cada paso del proceso de modelado.	68
5.2. Nube de puntos sintética. Porcentaje de puntos ajustados, distancia media al modelo y desviación estándar.	71
5.3. Nube de puntos del Castillo de Torrelobatón. Porcentaje de puntos ajus- tados, distancia media al modelo ajustado y desviación estándar.	72
6.1. Tabla salarial del convenio del metal.	79
6.2. Coste anual para la empresa de un ingeniero.	79
6.3. Distribución temporal de trabajo	80
6.4. Coste asociado al material.	81
6.5. Coste asociado a las aplicaciones software.	81
6.6. Material fungible	82
6.7. Costes indirectos	82
6.8. Coste total del proyecto	83

Abreviaturas

AIC	A rquitectura I ngeniería y C onstrucción
API	A pplication textbfProgramming I nterface
BIM	B uilding I nformation M odelling
CAD	C omputer A ided D esign
DAP	D ata A dquisition P rotocol
DEM	D igital E levation M odel
DLL	D ynamic L ink L ibrary
H-BIM	H eritage B uilding I nformation M odelling
HSV	H ue S aturation V alue
IFC	I ndustry F oundation C lasses
NURBS	N on- U niform R ational B - S pline
LOD	L evel O f D etail
PCL	P oint C loud L ibrary
PS	P hase S hift
RANSAC	R ANdom S Amples C onsensus
RGB	R ed G reen B lue
SDK	S oftware D evelopment K it
TOF	T ime O f F light

Capítulo 1

Introducción

En este capítulo se introducen los conceptos básicos sobre el tema tratado en el trabajo de final de máster y se presenta el objetivo perseguido: el desarrollo de una metodología que, utilizando algoritmos basados en técnicas de visión artificial, permita generar la reconstrucción virtual de una edificación de forma automática a partir de una nube de puntos. También se analiza el estado de la tecnología a través de un análisis crítico de los resultados más relevantes publicados en los últimos años tanto en revistas como congresos. Finalmente, se describe el caso de estudio utilizado para testar el desempeño de los algoritmos desarrollados.

1.1. Introducción

España atesora un patrimonio cultural sobresaliente en términos de calidad y cantidad. La diversidad de monumentos, cascos históricos, edificios religiosos, castillos o palacios es enorme y no sólo es parte de nuestra identidad cultural sino que genera un importante retorno económico que compensa en gran medida el esfuerzo realizado para su preservación. La digitalización de patrimonio debe verse como una estrategia para su conservación a largo plazo y las nuevas tecnologías de captura deben ser la base de este sofisticado y complejo proceso.

La aproximación actual para la documentación de patrimonio consiste en la adquisición, organización e integración de datos de construcciones históricas en un único proyecto empleando la metodología de trabajo Building Information Modelling (BIM). La metodología BIM está siendo adoptada ampliamente a nivel mundial y propone la creación y gestión de proyectos de construcción centralizando toda la información del proyecto en el mismo modelo digital, lo que supone una gran evolución respecto a los sistemas de diseño tradicionales basados en planos.

La generación de un modelo virtual es fundamental para trabajos posteriores tanto de conservación como de restauración. Es de destacar que la generación del modelo virtual es laboriosa y demanda el uso de una gran cantidad de tiempo. Por este motivo cada vez más compañías e investigadores dedican grandes esfuerzos al desarrollo de herramientas y metodologías que permitan generar de forma automática o semi-automática dicho modelo a partir de las digitalizaciones realizadas con el objetivo de disminuir el tiempo de modelado.

1.2. Objetivo

En el presente Trabajo Fin de Máster se aborda el desarrollo de una novedosa metodología para tratar de forma automática los datos tridimensionales capturados en forma de nube de puntos de la edificación estudiada. La metodología pretende definir los elementos básicos del modelo virtual sólido a partir de dicha nube sin que sea necesaria la intervención de un experto en ninguna de las fases del proceso.

La metodología propuesta contempla varias fases de procesamiento. Para cada una de estas fases se han desarrollado algoritmos que, utilizando técnicas de visión artificial, son capaces de detectar y modelar de forma automática los elementos arquitectónicos más representativos de edificaciones históricas. Concretamente, los algoritmos han sido optimizados para el análisis de los muros y las torres de edificaciones defensivas como los castillos, detectados aplicando técnicas de segmentación y etiquetado inteligente.

La metodología propuesta ha sido integrada, a través de los diferentes algoritmos desarrollados, en una herramienta informática. La herramienta, que ha sido desarrollada en forma de plug-in, puede ser utilizada directamente en el entorno BIM Revit de la compañía Autodesk, lo que mejora notablemente la usabilidad de los desarrollos propuestos.

1.3. Estado de la tecnología

En la actualidad existe una demanda real de sistemas que permitan la conversión automática de nubes de puntos en elementos BIM. Sin embargo, los sistemas de detección y parametrización automáticos se encuentran aún en fase de desarrollo y pertenecen principalmente al mundo académico. Este tipo de utilidades no están presentes o cuentan con un rango de aplicaciones muy limitado en aplicaciones comerciales de diseño y/o modelado debido a la dificultad inherente al procesamiento automático de las nubes.

Se debe notar que el proceso para transformar una nube en un modelo BIM no es automático ni viable y aún requiere de la intervención de un experto porque la inteligencia artificial aún no se ha desarrollado hasta el punto en el que sea capaz de realizar ciertas hipótesis basadas en criterios arquitectónicos por ella misma [22, 23, 26, 36, 40].

En el mundo académico se han documentado precedentes de sistemas semi-automáticos de extracción de características a partir de modelos 3D. Autores como Wang y otros [38] se limitan a la extracción de planos y sus intersecciones para definir paredes, suelos, techos, puertas y ventanas para edificaciones con geometrías muy sencillas (Figura 1.1).

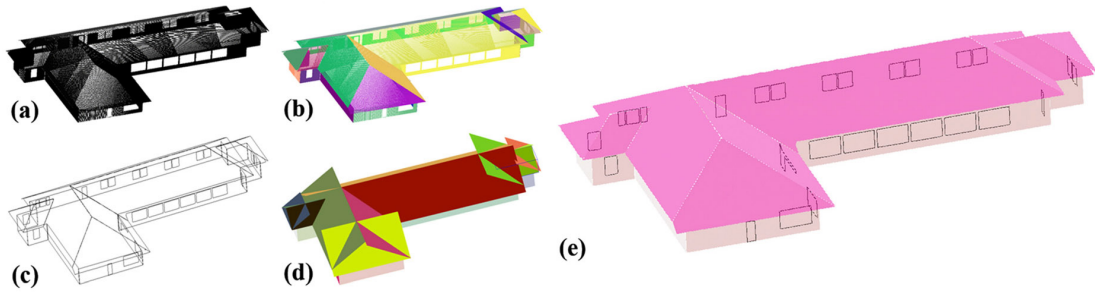


FIGURA 1.1: Wang y otros. Caso de estudio. a) Nube original b) Nube segmentada. c) Detección de límites y bordes d) Ajuste de geometría e) Modelo semántico final

Zolanvari y Laefer [41] detectan puertas y ventanas seccionando horizontal y verticalmente las nubes de puntos de fachadas de bloques de edificios residenciales (Figura 1.2). Aunque consiguen determinar los bordes correctamente, la metodología empleada sólo puede ser aplicada a fachadas sencillas y planas en las que se presupone la existencia de aberturas con formas regulares.

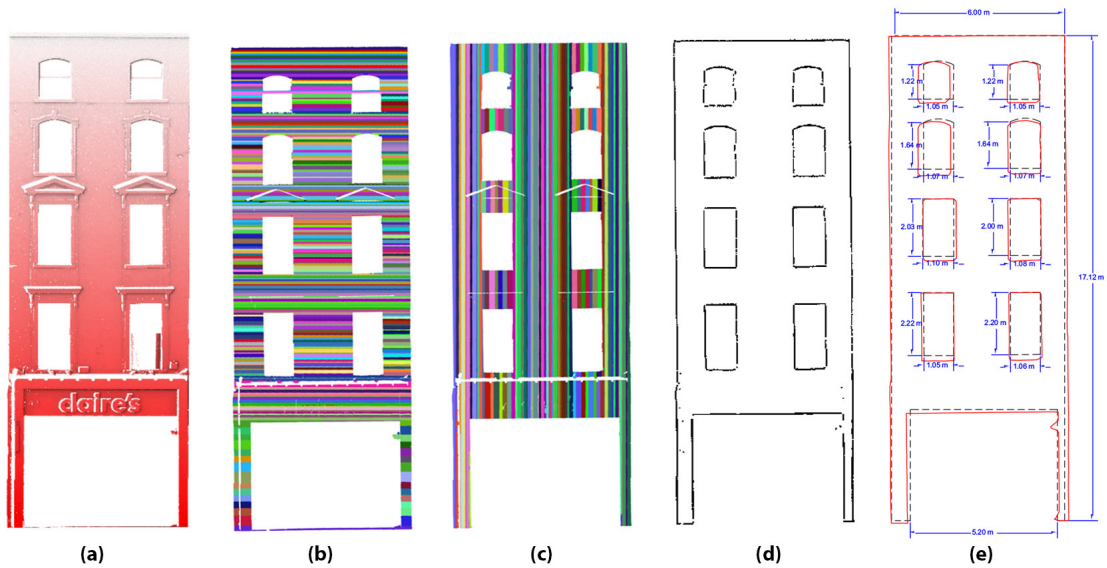


FIGURA 1.2: Zolanvari y Laefer. Caso de estudio. a) Nube original b) Segmentación horizontal c) Segmentación vertical d) Bordes detectados e) Dibujo con mediciones

Li y otros [19] realizan una segmentación y etiquetado basados en altura para definir volúmenes prismáticos a través de la intersección de planos en fachadas de edificios. Aunque modelan la fachada de una catedral como caso de estudio, las partes más complejas son directamente descartadas (Figura 1.3).

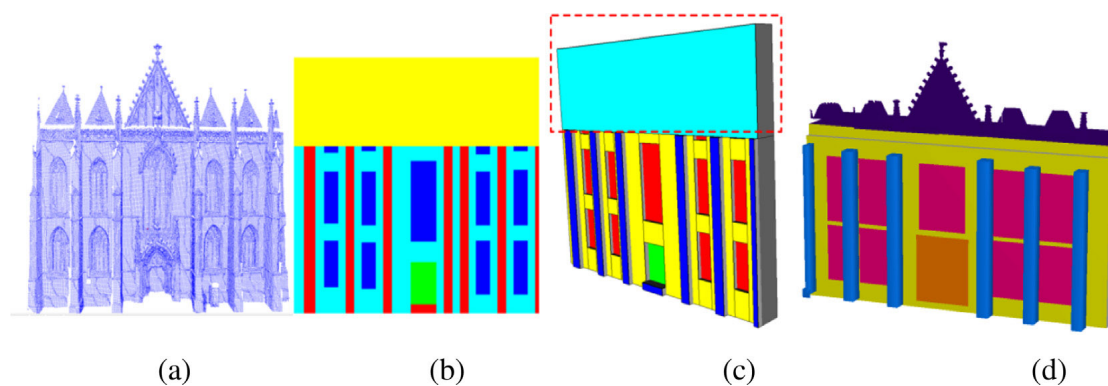


FIGURA 1.3: Li y otros. Caso de estudio. a) Nube de puntos b) Segmentación c) Reconstrucción 3D d) Modelo final

Stambler y otros [35] estiman la distribución en planta más probable de una edificación realizando hipótesis sobre la distribución de los muros (Figura 1.4). Su algoritmo optimiza el modelo contrastando las hipótesis realizadas para generar las habitaciones más probables.

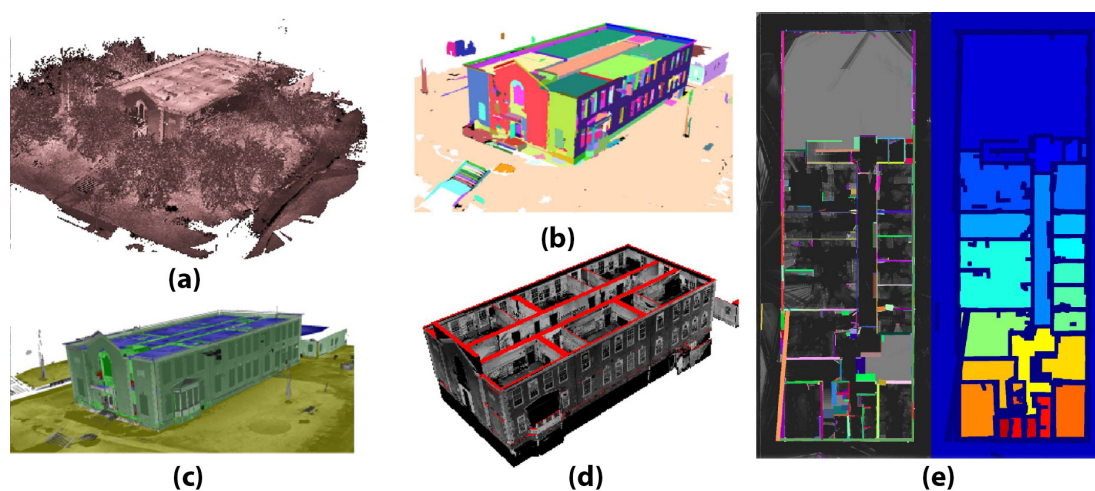


FIGURA 1.4: Stambler. Caso de estudio. a) Nube de puntos. b) Segmentación. c) Superficies coloreadas por clases. d) Volumen final con nube de puntos de intensidades. e) Plano en planta.

Ochman [28] sigue una aproximación similar al realizar la reconstrucción paramétrica del modelo a través del estudio de la conectividad de las paredes de forma global (Figura 1.5).

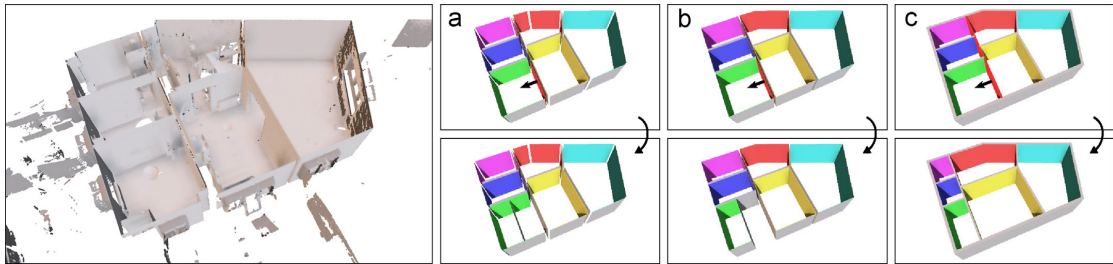


FIGURA 1.5: Ochmann. Caso de estudio.

Aunque ambos detectan correctamente la distribución de estancias se basan en hipótesis difícilmente aplicables a patrimonio como la poca irregularidad de la paredes, que en todo caso deben ser planas para poder definir correctamente las divisiones. Además, requieren del uso de nubes de puntos interiores del edificio estudiado, información que no está disponible para este caso de estudio.

Por otra parte, en el mercado existe un limitado número de plug-ins para el tratamiento de nubes de puntos directamente en entornos BIM. Estos plug-ins están concebidos para que el usuario pueda tener todas las ventajas de una aplicación de tratamiento de nubes de puntos de alta eficiencia directamente en estos entornos de trabajo, pero restringido a unos formatos específicos de nubes de puntos que dependen de la marca y modelo del escáner láser con que se generan las nubes de puntos. El plug-in CloudWorx[16] para Revit de la compañía LEICA es un ejemplo representativo y provee utilidades para la identificación automática de elementos como tuberías en obras de nueva edificación. En cualquier caso, este tipo de plug-in están solamente orientado a datos obtenidos mediante escaneado láser en edificación civil y no para patrimonio, por lo que no cumplen los requerimientos de este tipo de proyectos.

1.4. Caso de estudio

La metodología desarrollada permite detectar y parametrizar un conjunto básico de elementos arquitectónicos a partir de la nube de puntos de partida. Los elementos detectados pretenden conformar un proyecto básico de trabajo donde se puedan añadir elementos más complejos y detallados en una etapa posterior del proceso de reconstrucción digital. La metodología propuesta, cuyos algoritmos y funcionalidades han sido integrados en

un plug-in, permite trabajar directamente sobre el software Revit, un programa ampliamente utilizado por arquitectos e ingenieros para el desarrollo de proyectos BIM, lo que mejora significativamente la usabilidad del desarrollo realizado.

La metodología propuesta reconoce, parametriza y determina la interdependencia de los elementos arquitectónicos más representativos presentes en la nube estudiada como muros y torres que matemáticamente han sido estudiados como planos y cilindros. El proceso se realiza directamente en el programa de modelado y sin la intervención directa de un experto en ninguno de los pasos involucrados.

Para poder estudiar el desempeño de los diferentes algoritmos implementados y validar su comportamiento se ha contado con nubes de puntos que contienen elementos arquitectónicos representativos. En concreto, los algoritmos han sido testados empleando fragmentos de una nube de puntos del Castillo de los Comuneros de Torrelobatón, uno de los castillos mejor conservados de Castilla y León y declarado patrimonio histórico en 1949 (Figura 1.6). La nube original, propiedad de la Junta de Castilla y León, contiene los elementos más representativos de la arquitectura defensiva Medieval y Renacentista en Europa.



FIGURA 1.6: Digitalización del castillo de Torrelobatón

1.5. Antecedentes

El trabajo descrito en esta memoria se basa en algunos de los desarrollos realizados durante la participación del autor en el proyecto INCEPTION (Inclusive cultural heritage in Europe through 3D semantic modelling). El proyecto INCEPTION, fundado por el programa de investigación e innovación de la Unión Europea, tiene como fin la innovación en el modelado 3D aplicado a patrimonio cultural a través de la reconstrucción tridimensional dinámica de entornos edificados y sociales. Pretende enriquecer la identidad europea a través de la comprensión de cómo el patrimonio cultural europeo evoluciona a lo largo de períodos extensos de tiempo [13].

Uno de los desarrollos generados como parte del proyecto INCEPTION es la aplicación 3DASH. Esta aplicación emplea una versión inicial de algunos de los algoritmos descritos en esta memoria. Las novedosas características de la herramienta han permitido publicar un artículo breve de congreso titulado "Supporting the Automatic Extraction of HBIM Elements from Point Clouds" (Asistencia en la Extracción automática de elementos HBIM a partir de nubes de puntos) publicada en la 7^a conferencia internacional sobre digitalización de patrimonio EuroMed 2018 en el que se avanza brevemente el funcionamiento de la aplicación 3DASH (ver Román y otros [32]). Además, se cuenta con un segundo artículo de congreso, ya aceptado en el congreso CIPA 2019 [7] y en fase de revisión, titulado "Towards automatic 3D parametrization of non-planar surfaces" donde se detalla tanto el funcionamiento de los algoritmos desarrollados como los satisfactorios resultados obtenidos.

1.6. Estructura del documento

Esta memoria se ha estructurado en siete capítulos. El contenido de los apartados queda descrito a continuación:

Capítulo 2 Tecnologías de digitalización tridimensional existentes. Composición formal de las nubes de puntos y las mallas. Tecnologías de digitalización utilizadas en proyectos relacionados con patrimonio: fotomodelado y escaneado láser.

Capítulo 3 Entornos profesionales para diseño BIM. Importación nativa de nubes de puntos en Revit y ArchiCAD, ventajas e inconvenientes. Metodología tradicional de generación de modelos paramétricos en patrimonio.

Capítulo 4 Herramientas comerciales para el tratamiento de nubes de puntos. Bibliotecas software de tratamiento de nubes de puntos. Metodología de tratamiento de nubes de puntos desarrollada. Integración de los algoritmos en Revit.

Capítulo 5 Pruebas realizadas. Desempeño de los algoritmos.

Capítulo 6 Estudio teórico del coste aproximado de la realización del proyecto.

Capítulo 7 Logros técnicos conseguidos. Aplicaciones prácticas de la metodología. Conclusiones a las que se ha llegado tras analizar los resultados obtenidos. Futuras líneas de desarrollo.

Capítulo 2

Tecnologías de digitalización tridimensional

En este capítulo se hace una breve exposición de las tecnologías de digitalización tridimensional existentes y se analiza la composición formal de los conjuntos de datos (nubes de puntos y mallas) generados durante el proceso. Finalmente, se exponen y analizan las tecnologías de digitalización más ampliamente utilizadas en proyectos de documentación, conservación y restauración de patrimonio como son el fotomodelado y el escaneado láser.

Durante los últimos años la disponibilidad de técnicas analíticas y equipos no destructivos para el estudio de patrimonio cultural ha aumentado de manera notable. Los aparatos, cada vez más precisos, portátiles y accesibles, han propiciado que el uso de tecnologías de digitalización se hayan convertido en una herramienta muy importante para la documentación de bienes culturales [14, 30]. Algunas de las aplicaciones en las que este tipo de tecnologías han encontrado un nicho relevante de aplicación son, entre otras, las siguientes:

- Documentación del estado, evolución y contextualización histórica de bienes.
- Detección, documentación y seguimiento de la evolución de deterioros.
- Medición de distancias, superficies, volúmenes y ángulos.
- Generación de modelos.
- Monitorización y asesoramiento.
- Planificación de operaciones de protección, conservación y rehabilitación.

Además, el escaneado láser y el fotomodelado son hoy en día técnicas populares para adquirir datos en bruto sobre la geometría de edificaciones y elementos arquitectónicos, algo que se ha visto reforzado por mejoras significativas en el campo de la tecnología de escaneado y el procesamiento automático de imágenes. Muchos problemas que fueron catalogados en el pasado como críticos en el escaneado láser, como el costoso esfuerzo que supone la adquisición y el registrado, han dejado de ser a día de hoy un problema gracias al aumento de la automatización de estos procedimientos.

2.1. Clasificación de las técnicas de digitalización

Existen numerosas técnicas de digitalización 3D y una gran cantidad de criterios que se pueden aplicar para su clasificación. Generalmente, la distinción más básica se realiza en función del tipo de contacto que se da entre el aparato de medida y el objeto. Si el elemento de medida se apoya sobre la superficie la técnica se denomina de contacto. Estas técnicas son usadas en control dimensional de procesos de fabricación y cuentan con una elevada precisión, cercana a la centésima de milímetro. Sin embargo, las técnicas

de contacto son lentas y no permiten trabajar con objetos frágiles ya que el elemento de medida es una punta de acero duro o zafiro que puede dañar la superficie analizada, algo a tener especialmente en cuenta en digitalización de patrimonio.

En las técnicas sin contacto el elemento de medida no se apoya sobre la superficie medida y pueden ser clasificadas en técnicas de transmisión como la tomografía, los ultrasonidos o la resonancia magnética y técnicas de reflexión. Dentro de la categoría de técnicas de reflexión no ópticas se encuentran los sonares, que emplean ondas mecánicas. Por otra parte, las técnicas que usan reflexión óptica se dividen en dos tipos, técnicas de escaneado activo y de escaneado pasivo [10].

Las técnicas de escaneado pasivo no emiten radiación y utilizan la radiación ambiental para generar el modelo. Una gran parte de las técnicas de este tipo emplean luz del espectro visible aunque existen técnicas que utilizan otros espectros como el infrarrojo. Las técnicas pasivas, entre las que se encuentra el fotomodelado, son en general baratas porque no necesitan de elementos adicionales para la emisión de ondas electromagnéticas.

Finalmente, los técnicas activas emiten radiación que se dirige hacia el objeto a escanear. Estos sistemas utilizan todo tipo de longitudes de onda y abarcan desde las ondas de radio hasta rayos X. Cuando la onda reflejada por el objeto es captada por el sensor del sistema de medida se determina la geometría de la escena empleando diferentes principios físicos. Ejemplos representativos de sistemas activos son el escáner láser 3D, que determina la distancia al objeto utilizando un haz láser pulsado o el escáner de luz estructurada, que determina la forma del objeto mediante la proyección de un patrón de luz y su registro en un sistema. En la Figura 2.1 se recoge la clasificación presentada.

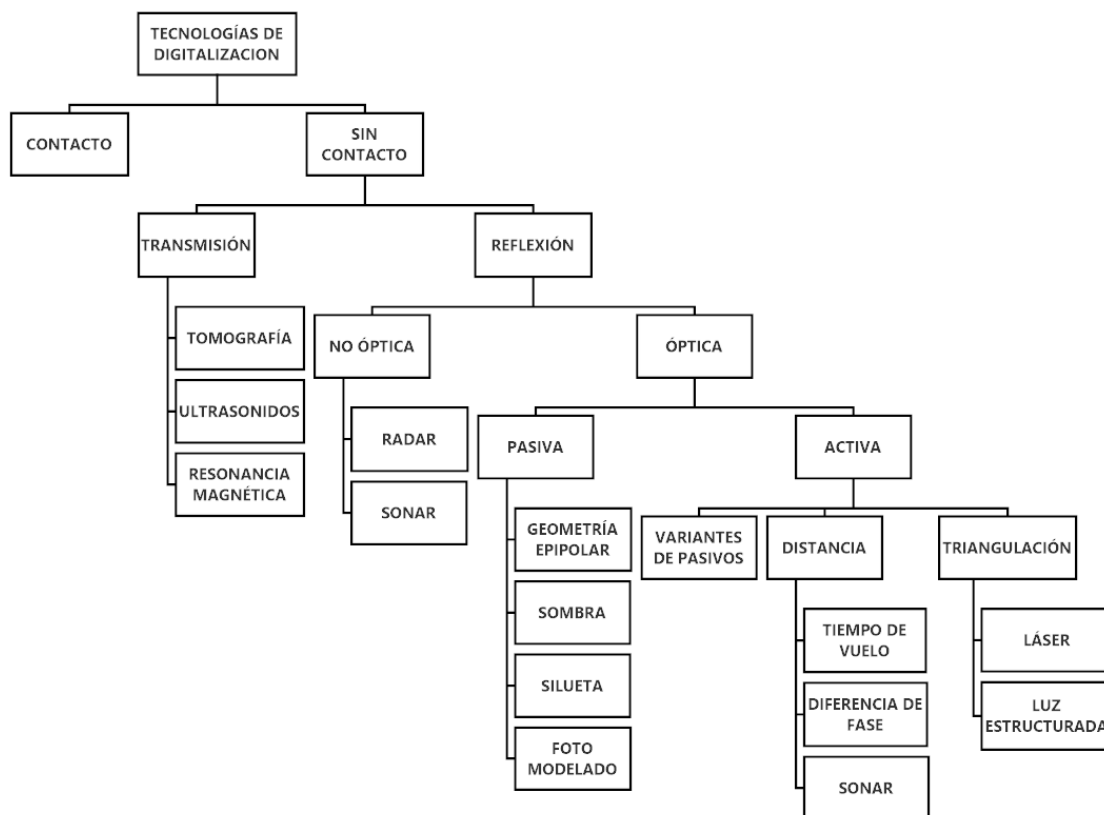


FIGURA 2.1: Clasificación de las tecnologías de digitalización más representativas.

2.2. Representación de modelos 3D

En todos los procesos de modelado mediante digitalización 3D los sistemas de captación determinan y almacenan puntos discretos que representan coordenadas geométricas de la superficie del modelo digitalizado. A estos conjuntos de datos se les denomina nubes de puntos y se corresponden con el primer conjunto de datos que se genera tras realizar un escaneo láser o generar un modelo mediante fotomodelado.

En una segunda etapa de procesamiento se puede obtener una malla triangular a partir de la nube de puntos empleando unos determinados algoritmos. La malla permite pasar de un modelo 3D formado únicamente por puntos a un modelo 3D formado por planos poligonales, generalmente triángulos cuyos vértices son los puntos capturados.

2.2.1. Nube de puntos

Las nubes de puntos están compuestas por vértices superficiales situados en un sistema de referencia tridimensional. Los vértices de la nube, que vienen definidos por el conjunto

$V = \{p_i \mid p_i \in \mathbb{R}^3, 1 \leq i \leq 3\}$, generalmente contienen las coordenadas cartesianas X , Y y Z de la superficie externa de los objetos registrados aunque algunos dispositivos permiten incorporar información adicional como:

- Información sobre el color (espacio RGB ó HSV).
- Información sobre la reflectividad.
- Información termográfica.

La estructura de los campos que almacenan los vértices son generalmente definidos en las cabeceras de los archivos que almacenan las nubes. El formato ply, uno de los más comunes en digitalización, define en la cabecera del archivo (Fragmento 2.1) la estructura de los campos con la directiva **property**.

La utilización de nubes de puntos ha experimentado un gran crecimiento a lo largo de los últimos años debido al gran desarrollo que han sufrido las tecnologías de digitalización y que ha propiciado que numerosas industrias hayan adoptado este tipo de tecnologías en sus procesos de diseño. En el sector AIC las nubes de puntos se utilizan para generar proyectos de construcción a modo de plantilla durante el proceso de diseño. También son utilizadas al diseñar operaciones de intervención sobre la edificación o para la realizar visitas virtuales del lugar.

```
1 ply
2 format ascii 1.0
3 element vertex 177397
4 property float x
5 property float y
6 property float z
7 property uchar red
8 property uchar green
9 property uchar blue
10 end_header
11 113.866 -147.679 143.719 38 47 45
12 113.324 -147.742 143.709 41 51 53
13 113.582 -147.79 143.716 39 47 47
```

LISTING 2.1: Cabecera de una nube en formato ply. Tres vértices de la nube.

2.2.2. Malla de triángulos

Las nubes pueden ser procesadas para obtener mallas de triángulos 3D que describen la superficie tanto en forma como en dimensiones. Para la generación de mallas existen distintos algoritmos que, empleando la información contenida en la nube, definen los diferentes triángulos de la malla 3D, siendo los vértices de los polígonos los puntos de la nube.

La técnica más habitual de generación de mallas utiliza el algoritmo de triangulación de Delaunay que calcula una red de triángulos que cumple la condición de Delaunay. Esta condición establece que la circunferencia circunscrita de cada triángulo no contiene ningún vértice de otro triángulo.

Formalmente, una malla tridimensional M está definida por una tupla $\{V, E, F\}$, donde los vértices son $V = \{p_i \mid p_i \in \mathbb{R}^3, 1 \leq i \leq m\}$, las aristas son $E = \{e_{ij} = (p_i, p_j) \mid p_i, p_j \in V, i \neq j\}$ y F son las caras, usualmente triángulos $F = \{f_{ijk} = (p_i, p_j, p_k) \mid p_i, p_j, p_k \in V, i \neq j, i \neq k, j \neq k\}$.

Las mallas se pueden clasificar según el método utilizado para almacenar la información. En el sistema vértice-vértice se indica para cada vértice la lista de vértices con los que está conectado. Esta representación es la más sencilla de implementar pero implica que se tiene que deducir la información de las aristas y las caras. Por otra parte, el sistema cara-vértice, compuesto por dos tablas, está formado por una tabla que contiene los vértices que componen cada cara (Fragmento 2.2: líneas con directiva `f`) y una segunda tabla (Fragmento 2.2: líneas con directiva `v`) en la que se indica las coordenadas de los vértices. Esta segunda representación es la más común.

Finalmente, a partir de las mallas es posible generar superficies, generalmente de tipo NURBS (Non-Uniform Rational B-Spline). Estas superficies se cosen generando modelos sólidos y que pueden ser interpretados por todos los sistemas CAD. La generación de curvas NURBS a partir de mallas queda fuera del ámbito de estudio de esta memoria.

```
1 # OBJ file format with ext .obj
2 # vertex count = 2503
3 # face count = 4968
4 v -3.4101800e-003 1.3031957e-001 2.1754370e-002
5 v -8.1719160e-002 1.5250145e-001 2.9656090e-002
6 v -3.0543480e-002 1.2477885e-001 1.0983400e-003
7 v -2.4901590e-002 1.1211138e-001 3.7560240e-002
8 v -1.8405680e-002 1.7843055e-001 -2.4219580e-002
9 f 1069 1647 1578
10 f 1058 909 939
11 f 421 1176 238
12 f 1055 1101 1042
13 f 238 1059 1126
14 f 1254 30 1261
```

LISTING 2.2: Ejemplo de malla en formato obj. Sistema cara-vértices.

2.3. Digitalización aplicada a patrimonio cultural

En proyectos de digitalización de patrimonio las escenas escaneadas suelen ser complejas y los métodos de medición puntual requieren de un alto nivel de detalle superficial para la generación del modelo. Además, en multitud de aplicaciones de este tipo es necesario contar con densas nubes de puntos, especificaciones que algunas de las tecnologías de digitalización no son capaces de cumplir.

De entre las diferentes técnicas de digitalización son dos las que en la actualidad han demostrado ser más adecuadas para la digitalización de patrimonio: el fotomodelado y el escaneo láser 3D. Cada una de estas dos tecnologías tienen sus propias ventajas e inconvenientes y se debe tener claro el campo de aplicación de cada una de ellas. También se debe tener en cuenta aspectos como el precio, fidelidad y calidad con la que se pueden obtener los modelos 3D utilizando estas tecnologías.

2.3.1. Escaneo 3D

Los escáneres láser 3D entran en la categoría de sistemas activos sin contacto y pueden ser clasificados en función de la tecnología que emplean: tiempo de vuelo (TOF) o desplazamiento de fase (PS). Los sistemas TOF emiten una única pulsación de la luz láser y determinan así la distancia hasta la superficie midiendo el tiempo necesario para que la luz reflejada regrese al sensor. De forma similar, los sistemas PS emiten una luz láser pero a una frecuencia específica. El ángulo de fase de la onda reflejada, tras impactar con

una superficie, es modificado. El desplazamiento de la fase de la onda es un indicador que permite calcular de forma precisa la distancia entre el punto de impacto y el dispositivo.

Ambas tecnologías están sujetas a una constante evolución, principalmente debido a los grandes avances tecnológicos que se han dado en los últimos años en las ópticas, el procesamiento de señal, el desarrollo de la electrónica y la mejora de los microprocesadores. Hasta hace unos pocos años las tecnologías TOF y PS podían ser difícilmente comparables, al menos bajo los mismos términos y requerimientos. TOF permitía mediciones precisas en largas distancias mientras que PS generaba mediciones con mayores niveles de ruido pero con una velocidad de adquisición muy elevada [1, 34]. Hoy en día ambas tecnologías parecen converger ya que el rango de medida de la tecnología PS ha crecido hasta los 350 metros con una reducción significativa del ruido en las medidas mientras que TOF ha incrementado la velocidad de captura hasta el millón de puntos por segundo manteniendo el nivel de precisión y el campo de visión ($360^{\circ}H \times 270^{\circ}V$). Por lo tanto, se puede concluir que ambas tecnologías son adecuadas para la digitalización de patrimonio.

2.3.2. Fotomodelado

El fotomodelado es una técnica de escaneo 3D pasiva. La luz ambiente reflejada por los objetos es capturada por una cámara desde dos o más lugares. Los algoritmos en que se basa el fotomodelado identifican y marcan diferentes puntos singulares que aparezcan en múltiples fotografías. Estas singularidades son automáticamente referenciadas en tres dimensiones y emparejadas para generar la nube de puntos.

El fotomodelado tiene sus orígenes en dos campos:

- La creación mediante fotogrametría de Modelos de Elevaciones Digitales (Digital Elevation Model) en topografía. Originalmente, los DEMs se creaban a mano, punto a punto, empleando un dispositivo conocido como estero-ploter, que solapaba imágenes tomadas desde un avión. En la década de los 80 este proceso comienza a automatizarse con el uso de cinta de escaneo y técnicas modernas de procesamiento de imágenes.
- La visión artificial estereoscópica. En la década de los 70 se desarrollan algoritmos que simulan el mecanismo que utilizamos los humanos para detectar la profundidad. El ojo humano utiliza las imágenes 2D proyectadas sobre la retina para

reconstruir la escena tridimensional consiguiendo, de esta forma, que el observador sea consciente de la profundidad del objeto visualizado.

Según Walford y otros [37], desde 2008 debido al aumento de la capacidad de cálculo de los ordenadores y la generalización de la fotografía digital se ha desarrollado un sistema de fotomodelado de corto alcance, que está basado en los dos puntos anteriores y que lo hacen significativamente diferente de la tradicional fotogrametría:

- Núcleo fotogramétrico: se encarga de la calibración de la cámara y permite resolver de forma precisa la posición y orientación de ésta a partir de las fotos tomadas.
- Núcleo de procesamiento de imágenes estereoscópicas: el software de procesamiento de imágenes compara, a partir de dos fotos diferentes, diferentes regiones buscando concordancias entre ambas. Cuando se encuentran regiones con una relación óptima, la posición y orientación de las fotos, previamente calculada, se utiliza para calcular la posición tridimensional de esta región.

Para que se pueda realizar el proceso es necesario que la superficie escaneada tenga textura. Las superficies con una textura homogénea, como un trozo de papel o una chapa metálica no pueden ser digitalizadas porque el proceso de casado de regiones no puede ser resuelto. Por ello, es recomendable que la superficie tenga un patrón aleatorio.

Los programas de modelado mediante fotogrametría permiten exportar los resultados en diferentes sistemas de representación dependiendo de los requerimientos del usuario:

- Nubes de puntos densas y espaciadas.
- Mallas 3D.
- Datos de calibración y orientación de la cámara
- Ortofotos.

Las digitalizaciones deben ser escaladas una vez que se ha obtenido el modelo final. Es necesario contar con distancias de referencia que pueden ser tomadas a partir de un mapa pre-existente, representaciones cartográficas o tomando medidas directamente en la escena escaneada. Aunque las fotos de los patrones de medida pueden no ser útiles

para la reconstrucción del modelo permanecerán como anotaciones en la documentación de la captura. La escala del modelo se establece comparando la distancia entre puntos concretos de la representación digital con las medidas reales. De forma adicional, si los resultados 3D están espacialmente referenciados tendrán que ser orientados respecto de un sistema de coordenadas adecuado.

2.3.3. Comparación de tecnologías

2.3.3.1. Precisión y resolución

La precisión del escaneado láser depende de numerosos factores como la marca, el modelo o la última calibración realizada, pero generalmente los factores limitantes son la precisión de la electrónica y la resolución de los encoders de medida. Aunque hay mucha variación entre fabricantes y modelos, un ejemplo representativo puede ser una precisión de 5mm a 50m con un escáner montado en un trípode. Por otra parte, una precisión axial de 3.4cm a 24m de distancia es un ejemplo representativo para un edificio histórico cuando se emplea fotomodelado. Sin embargo, es posible alcanzar una precisión por debajo de 1cm utilizando cámaras de alta resolución.

Para una superficie y una distancia a la cámara dadas existe una resolución concreta que permite que el fotomodelado optimice los puntos generados. La resolución depende del algoritmo implementado en el software de procesamiento. Por supuesto, la textura de la superficie no puede ser modificada pero la resolución puede ser mejorada tomando imágenes más cercanas a la superficie o utilizando cámaras de alta resolución.

Para una superficie con una textura adecuada una muestra 3D por cada pixel de la imagen es generalmente suficiente. Por ejemplo, situar una cámara de 10MPx a 1m del objeto implica que cada pixel se corresponde aproximadamente con 0.3mm de superficie (axial). En este caso la resolución del muestreo 3D podría ser aproximadamente 0.3mm. Colocando la cámara a 10m del objeto la resolución de la nube de puntos disminuiría a 3mm.

La resolución de un escáner láser está limitada mecánicamente (ángulos de los encoders del proyector láser). La resolución depende del ángulo y por lo tanto, cuanto más lejano el objetivo menor será la resolución de la superficie. Una resolución media típica para

edificaciones históricas es 3cm a 7m [25], pero varía significativamente en función de las necesidades del reconocimiento. Por lo tanto, la resolución de las cámaras empleadas para fotomodelado puede llegar a ser similar a la resolución angular de un escáner láser. Por otra parte, un escáner láser no se ve afectado por la textura de la superficie, a diferencia de lo que ocurre con el fotomodelado.

2.3.3.2. Rango de medida

El rango de medida de los escáneres láser está relacionado con el tamaño del objeto a escanear: cuanto más lejos se sitúa el escáner mayor puede ser el objeto escaneado. Los escáneres montados en trípodes tienen habitualmente un rango de medida de entre 0.5m y 200m.

El rango de medida del fotomodelado 3D depende únicamente de las imágenes tomadas. Un elemento puede ser digitalizado si se puede tomar una fotografía del mismo, lo que otorga a esta tecnología el rango de uso más amplio posible que va desde la digitalización de una pelota hasta una montaña.

2.3.3.3. Tiempos

El tiempo de digitalizado se ve afectado principalmente por tres motivos:

- El tiempo para configurar el equipo.
- El tiempo para escanear el objeto.
- El tiempo para procesar los datos.

El tiempo para configurar el equipo láser depende de la aplicación, las condiciones del entorno y el propio escáner. Para aplicaciones en campo, como en digitalización de inmuebles históricos, la configuración del escáner puede llevar un periodo prolongado de tiempo. Para el fotomodelado sólo hay que encender la cámara y comprobar los ajustes. Además, cuando se realiza un escaneado láser desde más de una posición, el instrumento tiene que ser configurado de nuevo en cada nueva posición y se debe determinar la posición relativa del escáner respecto del resto de posiciones anteriores. Para el fotomodelado

se establece un único sistema de coordenadas para todas las imágenes durante la etapa de procesamiento, por lo que se pueden tomar imágenes desde múltiples localizaciones sin trabajo adicional.

El tiempo de escaneado principalmente depende, en los escáneres láser, de los ajustes que se realicen en el rango de escaneado, la resolución geométrica/cromática requerida y el número de posiciones de escaneado. La distancia al objeto y las condiciones climatológicas no afectan al tiempo pero sí a la calidad de los datos. Con fotomodelado sólo es necesario tomar las fotos del área estudiada en las mejores condiciones posibles.

Respecto al tiempo de procesamiento, el escaneado láser generalmente produce los puntos de la nube en tiempo real o requiere un mínimo procesamiento. Por otra parte, el fotomodelado requiere de un complejo procesamiento posterior y necesita alrededor de un minuto para producir entre 1.000 y 5.000 puntos de la nube en un ordenador moderno. Si se han realizado escaneados desde múltiples localizaciones el escaneado láser necesitará algo de post-procesamiento para generar un sistema de coordenadas consistente y unir las diferentes nubes de puntos. Este paso no es necesario utilizando fotomodelado.

2.3.3.4. Precio

El escaneado tridimensional tiene principalmente dos costes asociados:

- Coste de adquisición.
- Coste de mantenimiento.

La mayor parte de los escáneres, al ser equipos de precisión, requieren de mantenimiento periódico y calibración que depende de las condiciones de uso. El fotomodelado sólo requiere del mantenimiento típico de una cámara digital (generalmente poco o ninguno). Además, la calibración de los sistemas de fotomodelado puede ser realizada fácilmente y el instrumento no tiene que ser enviado al fabricante periódicamente como en el caso de un escáner láser.

Un sistema de fotomodelado completo, compuesto por una cámara digital estándar y el software correspondiente tiene un coste reducido. La cámara y el software juntos generalmente cuestan menos de 3.000 €. Un escáner láser puede costar decenas o incluso cientos

de miles de euros incluyendo la compra de licencias para la adquisición y procesamiento de datos. Por otra parte, el software para fotomodelado suele integrar herramientas para el tratamiento de nubes y existen diferentes soluciones gratuitas como Insight3D o PhotoTourism.

2.4. El castillo de los Comuneros de Torrelobatón

El modelo digitalizado del Castillo de Torrelobatón está compuesto por una nube de puntos y una malla de triángulos 3D asociada. La nube de puntos, utilizada para probar la metodología propuesta, contiene 890.000 vértices con color RGB asociado y ha sido generada mediante fotomodelado. La nube de puntos, propiedad de la Junta de Castilla y León, ha sido seleccionada por contener elementos representativos de la arquitectura defensiva de la Edad Media y el Renacimiento europea. Las características principales del proceso de digitalización están reflejadas en la Tabla 2.1.

DAP	Escala	LOD	Tolerancia	Resolución
A+	1:100	5 cm	2.5 cm	1.3 cm

TABLA 2.1: Características de la nube de puntos del Castillo de Torrelobatón

El parámetro DAP (*Protocolo para la Adquisición de Datos*) hace referencia al protocolo seguido para la digitalización de la escena [24]. El protocolo empleado contempla ocho pasos y para cada uno de ellos se definen requerimientos e indicadores específicos. Cada uno de los pasos puede ser evaluado en función del modo en que éste es abordado y la suma de los indicadores parciales da lugar a una puntuación total. El DAP establece una clasificación dividida en cuatro categorías incrementales: B, A, A+ y A++.

La categoría A se emplea en documentación y la morfología de la escena escaneada tiene una relevancia similar a la calidad de las mediciones realizadas. Por otra parte, la categoría A+ se emplea principalmente para preservación porque los estudios que cumplen este estándar requieren mediciones con una precisión y exactitud extremas.

El parámetro LOD (Level Of Detail) hace referencia a la complejidad de la representación del modelo 3D en función de la distancia a la que se encuentra ésta del espectador. En modelos arquitectónicos, el LOD indica cómo se han mapeado las características del mundo real y en qué medida el modelo se ajusta a su contraparte del mundo real.

Finalmente, los parámetros tolerancia y resolución hacen referencia respectivamente al error máximo de la medición y a la densidad de la nube generada.

Capítulo 3

Entorno de trabajo BIM

En este capítulo se hace un breve repaso de los entornos de trabajo BIM existentes en el mercado y usados más ampliamente a nivel internacional. Además, se explica el sistema de importación nativo de nubes de puntos de las dos aplicaciones más relevantes, Revit y ArchiCAD, y sus principales características, ventajas e inconvenientes. Finalmente, se expone la metodología que tradicionalmente los profesionales del sector AIC han utilizado para generar modelos sólidos paramétricos en proyectos arquitectónicos relacionados con patrimonio.

3.1. BIM en proyectos de patrimonio

En la actualidad un porcentaje relevante de los proyectos arquitectónicos que son llevados a cabo están relacionados con mejoras o remodelaciones de edificaciones entre los que se encuentran los relacionados con la preservación de patrimonio. Este tipo de proyectos presentan principalmente dos problemas asociados:

- Es necesario realizar un modelo con muchos elementos arquitectónicos.
- El proceso para integrar el complejo arquitectónico preexistente en el proceso BIM no es simple.

Se debe destacar que la información disponible acerca de la geometría de edificios históricos está generalmente poco organizada, no es precisa ni actualizada o incluso es inexistente [2, 15, 39]. La generación de un modelo sólido geométrico es fundamental para trabajos posteriores: definir el plan de conservación, planificar trabajos de restauración, etc. Por lo tanto, en la mayor parte de este tipo de proyectos es preciso digitalizar tridimensionalmente el modelo real para poder desarrollar un modelo virtual suficientemente exacto en el entorno BIM.

En particular, el proceso denominado *Escaneado a BIM* ha recibido mucha atención por parte de los expertos ya que las tecnologías de digitalización han demostrado ser durante muchos años más que adecuadas para esta tarea [4, 5, 12]. Sin embargo, hoy en día los datos, aunque obtenidos en 3D, siguen siendo utilizados por arquitectos e ingenieros en forma de documentos 2D. Existen varios factores que han retrasado la adopción de una metodología de trabajo más actualizada:

- La restauración sigue siendo la principal tarea realizada para asegurar la sostenibilidad de patrimonio.
- Se tiene un entendimiento limitado de los datos de las nubes de puntos así como de la capacidad para manejar grandes conjuntos de datos.
- Limitaciones de los programas para manejar nubes de puntos y modelos simultáneamente.

Cada vez más profesionales del sector se están dando cuenta de los beneficios que supone emplear la metodología BIM mientras que el hardware y el software necesarios para realizar este tipo de proyectos están en continuo desarrollo. Ambos motivos han dado lugar a que se tenga cada vez una mayor comprensión de las herramientas que trabajan bajo la filosofía BIM lo que ha propiciado que haya un mayor número de aplicaciones disponibles en el mercado. Además, se debe tener en cuenta que los proyectos de restauración de patrimonio son interdisciplinarios y que para que los diferentes expertos puedan trabajar juntos es necesario contar con una plataforma BIM de trabajo colaborativo.

Por otra parte, las tecnologías de modelado 3D, a pesar de tener un cierto recorrido, están sujetas a una continua evolución. En el campo de la digitalización de edificios históricos la aplicación de este tipo de técnicas no es tan habitual como cabría esperar, por lo que el desarrollo de los procesos de *Escaneado a BIM* está cobrando una gran importancia en los últimos años.

La aproximación actual para la documentación y el modelado de patrimonio consiste en recoger, organizar e integrar datos sobre las edificaciones estudiadas en una única estructura gráfico-semántica utilizando herramientas que trabajan bajo la filosofía BIM. Pasar de nubes de puntos a elementos BIM sigue siendo un problema clave que debe ser abordado en cada nuevo proyecto [11] y hoy en día existen dos formas principales para generar modelos útiles a partir de estos conjuntos de datos:

- Utilizar software de generación de elementos paramétricos a partir de las nubes de puntos que puedan ser importados en entornos BIM a través de formatos interoperables.
- Modelar directamente en software BIM a través de funciones especiales o aplicaciones complementarias de tratamiento de nubes de puntos.

La existencia de estas dos posibilidades revela el gran potencial que tienen las nubes de puntos como material de partida para aplicaciones BIM. Sin embargo, modelar directamente elementos BIM en aplicaciones que trabajen de forma nativa con esta metodología simplifica el flujo de trabajo ya que no es necesario emplear software de terceros. Además, utilizar esta estrategia permite utilizar todas las características nativas de la aplicación como las bibliotecas de elementos predefinidos.

Se debe notar que la completa automatización del proceso de conversión "Nube de puntos-BIM" no es actualmente viable y aún requiere de la intervención de un experto, ya que la inteligencia artificial no ha sido desarrollado hasta tal punto que pueda realizar hipótesis basadas en criterios arquitectónicos por su cuenta [3, 6, 20, 21, 36].

3.2. Herramientas comerciales para el modelado BIM de patrimonio

Las dos herramientas comerciales más famosas para BIM, Revit y ArchiCAD, permiten importar de forma nativa nubes de puntos en sus proyectos. Además, cabe destacar que los fabricantes de escáneres láser suelen proporcionar plug-ins que permiten utilizar de forma eficiente los datos de la nube dentro de estas aplicaciones para impulsar el modelado de edificios existentes. En cualquier caso, contar con este tipo de información es útil en una gran variedad de actividades relacionadas con BIM como rehabilitaciones, operaciones de intervención, gestionar el ciclo de vida del edificio y realizar visitas virtuales de emplazamientos de interés.

3.2.1. Importación de nubes en Revit

Las nubes de puntos pueden ser directamente importadas en Revit utilizando una amplia variedad de formatos. Una vez que se selecciona la nube, el programa lanza automáticamente un proceso de indexado para una correcta utilización del escaneado dentro de la aplicación. Las nubes de puntos originales se convierten a formatos indexados RCP y RCS (Figura 3.1). Las nubes de puntos se importan en formato RCS mientras que los archivos RCP son proyectos que engloban uno o varios escaneados RCS.

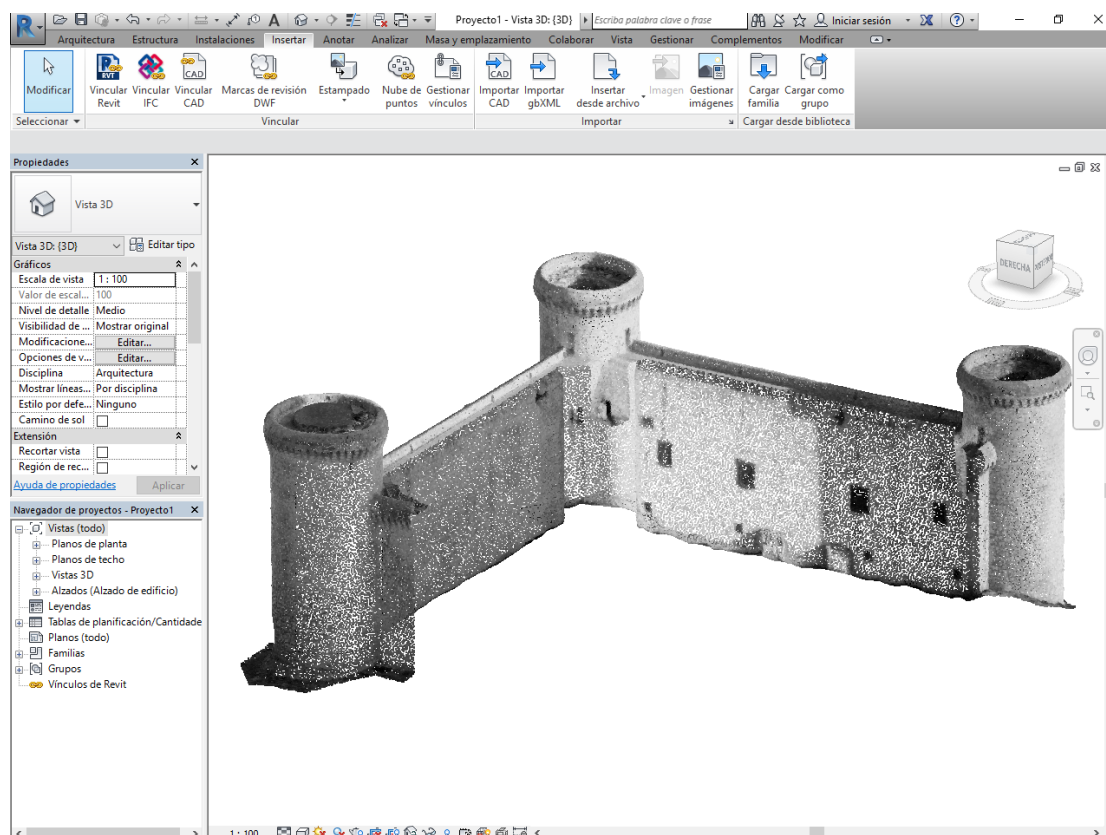


FIGURA 3.1: Nube de puntos importada en Revit

Si es necesario realizar conversiones de escaneados avanzadas, editar o visualizar las nubes de puntos es preferible utilizar el software Autodesk Recap antes de importar las nubes en Revit. Recap permite trabajar con nubes fácilmente y cuenta con opciones sencillas de exportación y de integración en no sólo toda la amplia gama del software de Autodesk, sino también en aplicaciones de terceros.

Se debe destacar que el formato E57 es el único estándar que permite trabajar con nubes de puntos independientemente del software BIM elegido. Aunque los datos en formato E57 son interoperables entre prácticamente todas las aplicaciones comerciales disponibles en el mercado, el intercambio de información es relativamente puro en el sentido de que no contiene más que información geométrica. Por su parte, Revit permite importar el formato E57 a través del software complementario Recap.

3.2.2. Importación de nubes en ArchiCAD

ArchiCAD permite importar nubes de puntos de forma nativa. Sin embargo, y al contrario que Revit, este software está restringido a formatos XYZ y E57. Además, ArchiCAD

impone que los puntos de las nubes importadas cuenten con seis campos (las tres coordenadas XYZ y los tres canales de color RGB). Esta característica implica que una nube en formato XYZ cuyos puntos cuenten tanto con tres como con cuatro campos (XYZ + reflectividad) deba pasar por una etapa de conversión intermedia. La conversión también se aplica a los archivos E57 que no cumplan con las condiciones mencionadas.

Para una conversión avanzada entre formatos es recomendable utilizar el software multiplataforma y de código abierto CloudCompare con el que se pueden editar archivos en formato XYZ y generar archivos E57 con un número adecuado de columnas. Con independencia de la conversión realizada, solo las coordenadas geométricas son importadas utilizando este procedimiento y el índice de reflectividad es desestimado. Como los campos RGB generados tras la conversión de la nube no contienen información acerca del color, la nube resultante es completamente negra lo que hace que la interpretación visual de las formas se vea seriamente comprometida. Esto implica que hay principalmente dos aproximaciones para trabajar en ArchiCAD con nubes de puntos no coloreadas:

- Capturar los datos de color de la escena utilizando imágenes 360°, lo que aumenta enormemente el tiempo de adquisición.
- Emplear una metodología que permita colorear la nube a partir de la reflectividad.

El segundo sistema es más adecuado ya que emplea información recogida durante la realización del escaneado. Una vez que la nube de puntos ha sido importada en ArchiCAD generalmente se posicionan los diferentes niveles del edificio respecto del suelo (Figura 3.2).

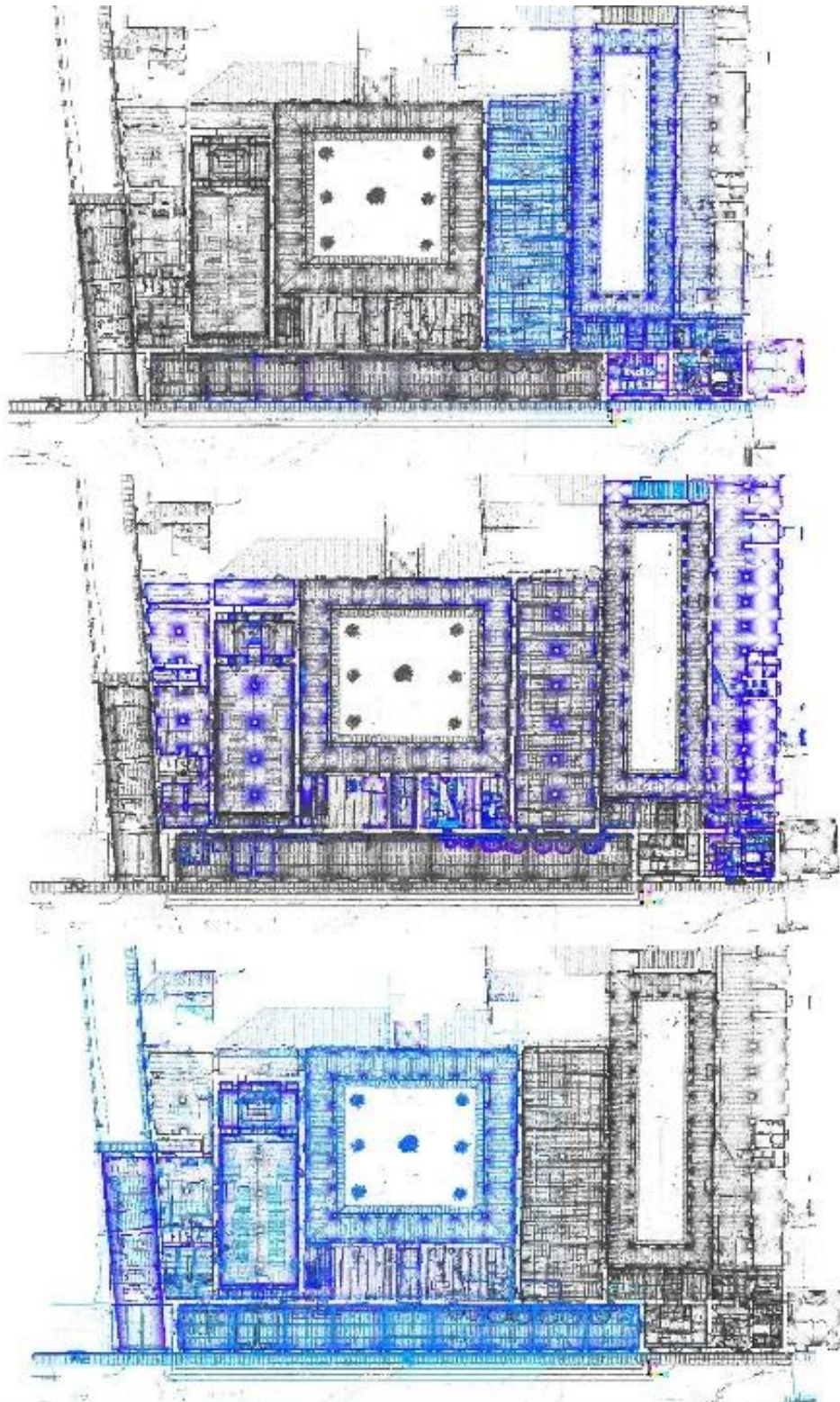


FIGURA 3.2: Vista en planta de la nube de puntos del Hospital de los Inocentes, Florencia (Italia), en ArchiCAD. (Fuente: Lerones y otros [18]).

3.3. Metodología de modelado con herramientas comerciales

Tradicionalmente, una vez que se importa la nube en el entorno BIM, ya sea Revit o ArchiCAD, los diseñadores generan el modelo sólido de la edificación empleando la nube, que usan a modo de plantilla. Generalmente, en una primera etapa de diseño se generan los niveles y las referencias más importantes de la edificación. Una vez concluida esta fase, una forma bastante común de abordar la generación del modelo es la creación de elementos arquitectónicos simples e independientes. Estas geometrías pueden ser generadas utilizando bibliotecas de elementos simples pre-existentes o bien ser modeladas in-situ en caso de que la geometría sea demasiado compleja como para ser parte de un estándar. Una vez que se han modelado los diferentes elementos se genera el modelo 3D completo de la edificación (Figura 3.3).

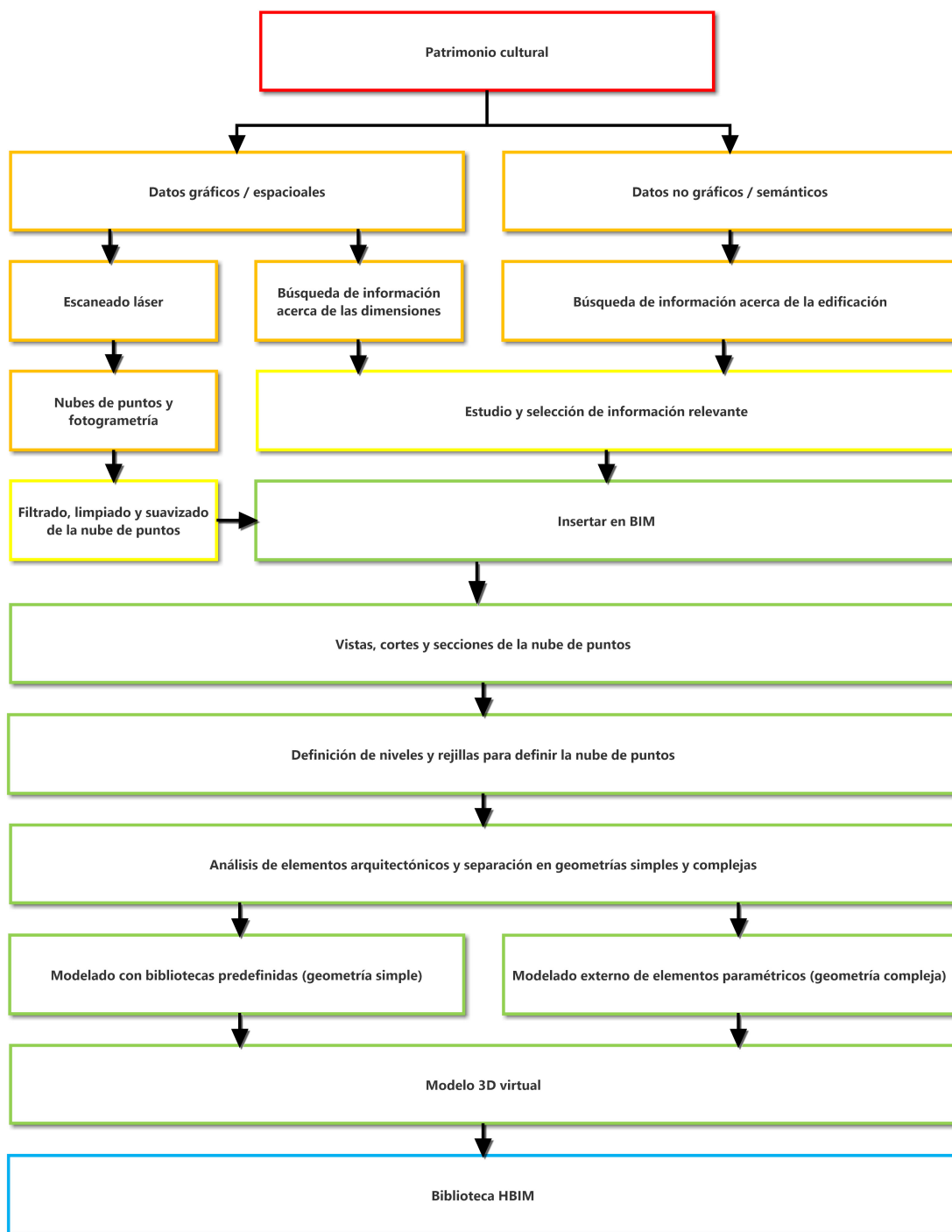


FIGURA 3.3: Diagrama de bloques del proceso seguido para realizar un modelo 3D y a partir de una nube de puntos. Naranja: primer paso; amarillo: segundo paso; Verde: tercer paso; Azul: resultados. (Fuente: López y otros [20]).

Se debe notar que mientras que las nubes de puntos son objetos indivisibles, cada uno de los puntos puede servir como una referencia para delinear las diferentes geometrías y por lo tanto, modelar elementos de la edificación a través de una interpretación crítica de los datos de la nube.

El modelo BIM se genera utilizando la nube de puntos post-procesada siguiendo unos ciertos criterios y requisitos tanto técnicos como arquitectónicos para garantizar que el modelo pueda ser transferido correctamente a un formato estandarizado e interoperable como es el IFC (Industry Foundation Classes). Al seguir estas recomendaciones se garantiza que todos los entornos de trabajo que operen con el estándar IFC sean capaces de acceder a la información contenida en dicho modelo.

El formato IFC es un formato de intercambio de datos que facilita las tareas de exportación e importación de información sin pérdida o distorsión de los datos. Es un formato abierto, neutro y no controlado por los desarrolladores de software que ha sido desarrollado para facilitar la interoperabilidad entre éstos.

En la Figura 3.4 se ha representado a modo de ejemplo la nube de puntos de la Iglesia de San Nicolás, Obergum (Países Bajos) mientras que en la Figura 3.5 se ha representado el modelo BIM generado en ArchiCAD a partir de dicha nube de puntos. El modelo BIM puede ser exportado en formato IFC, lo que permite que sea importado en otras aplicaciones como Revit. En cualquier caso, los modelos generados utilizando esta aproximación son costosos y requieren de un uso intensivo de tiempo y personal durante su desarrollo, lo que refuerza la necesidad de contar con metodologías que automaticen al máximo la generación de los modelos 3D.



FIGURA 3.4: Nube de puntos de la Iglesia de San Nicolás, Obergum (Países Bajos). (Fuente: Leronés y otros [18]).

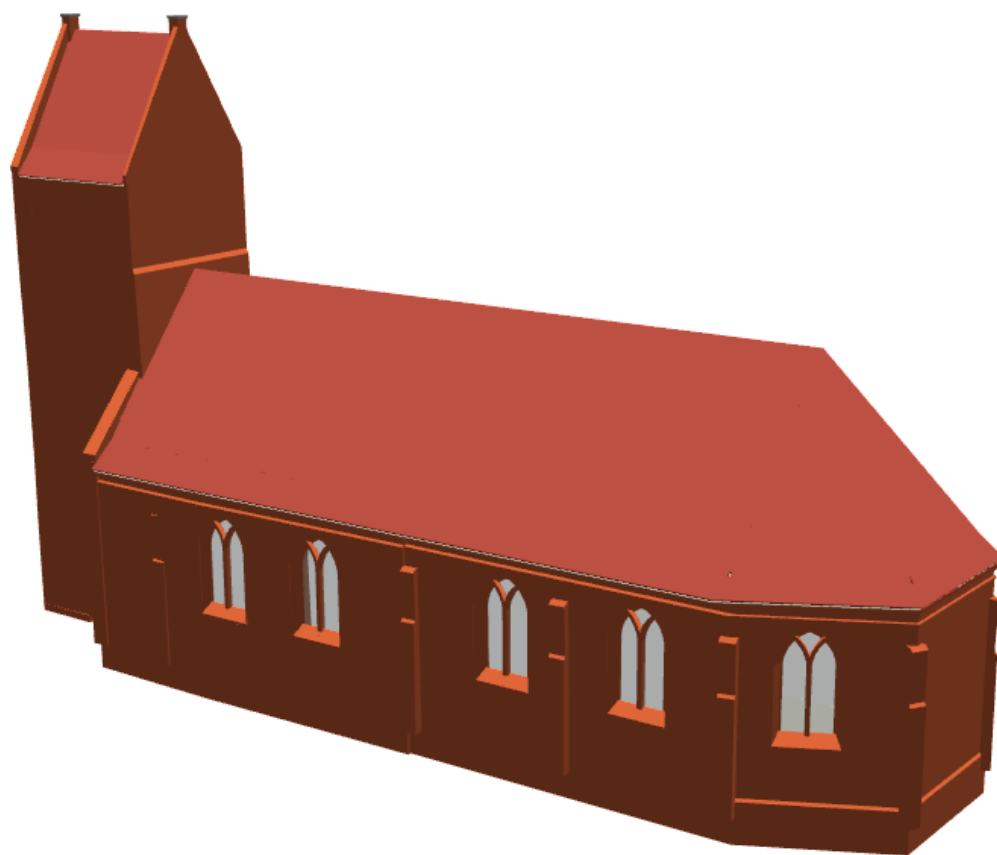


FIGURA 3.5: Modelo BIM de la Iglesia de San Nicolás en ArchiCAD y exportado a formato IFC2x3. (Fuente: Leronés y otros [18]).

Capítulo 4

Metodología de modelado automático

En este capítulo se hace un breve repaso de las herramientas comerciales que existen para el tratamiento de nubes de puntos así como de las bibliotecas software de tratamiento de estos conjuntos de datos. Además, se expone la nueva metodología de tratamiento de nubes de puntos desarrollada, que contempla desde la captura inicial de los datos hasta la obtención de un modelo sólido siguiendo la filosofía de trabajo BIM. Finalmente, se expone el sistema seguido para integrar los desarrollos realizados en un entorno BIM profesional como es Revit.

4.1. Tratamiento de nubes de puntos

4.1.1. Herramientas comerciales

Existen diferentes herramientas tanto gratuitas como de pago en el mercado para el tratamiento de nubes de puntos. Este tipo de software generalmente incluye opciones básicas para el mallado o triangulación, filtrado, reparación de huecos y contorneado. A continuación se indican algunas de las herramientas más representativas:

- MeshLab: Es un programa de procesamiento y edición de nubes de puntos y mallas libre y gratuito. Ofrece un conjunto de herramientas para la edición, limpieza, reparación, inspección, renderizado, texturizado y conversión de mallas. Además, ofrece características para el procesamiento de datos obtenidos mediante dispositivos de digitalización 3D y para la preparación de modelos imprimibles en 3D [27].
- CloudCompare: Es un programa gratuito y libre de procesamiento de nubes de puntos y de mallas. Originalmente fue diseñado para realizar comparaciones entre nubes de puntos densas o entre nubes de puntos y mallas. El software fue mejorado para que permitiera un procesamiento de nubes de puntos más genérico que incluyera características avanzadas como algoritmos de registro, remuestreo, manejo de color y normales, computación estadística, segmentación, gestión de sensores, segmentación interactiva o automática o mejora de visualizaciones [8].
- ReCap: Es un software para proyectos complejos de escaneado y fotomodelado. Permite exportar la nube de puntos generada a un formato propietario que puede ser importado con facilidad por otras aplicaciones de Autodesk como AutoCAD, Revit, Inventor o NaviWorks, entre otras [31].
- Cyclone: Software de la compañía Leica Geosystems. Es el programa de procesamiento de nubes líder del mercado. Cuenta con una familia de módulos de software que proporciona la mayor cantidad de opciones de procesos de trabajo para proyectos de escaneo láser 3D en aplicaciones de ingeniería, construcción, topográficas y similares [9].

Los paquetes de software mencionados permiten el tratamiento manual de nubes de puntos. Al ser programas de procesamiento genérico ninguno permite realizar un tratamiento

orientado al análisis e interpretación de nubes o si lo realizan, sólo son capaces de realizar un análisis básico como en el caso de CloudCompare, que es capaz de ajustar ciertas primitivas matemáticas a la nube.

Para desarrollar una aplicación de tratamiento de nubes a medida es necesario recurrir a soluciones más avanzadas. En este caso se ha optado por desarrollar una nueva metodología de detección y análisis de nubes de puntos orientada a edificaciones históricas que utiliza como motor de cálculo la biblioteca de visión artificial Point Cloud Library (PCL)[33].

4.1.2. Biblioteca PCL

La biblioteca de tratamiento de nubes de puntos PCL es capaz de manejar este conjunto de datos de forma eficiente y ha sido utilizada para dar soporte y aumentar la usabilidad y robustez del desarrollo realizado. La biblioteca es libre y gratuita y permite el tratamiento de imágenes 2D/3D y el procesamiento de nubes de puntos. La biblioteca contiene un amplio abanico de algoritmos novedosos de filtrado, estimación de características, reconstrucción de superficies, registrado, estimación de modelos y segmentación. Estos algoritmos pueden ser usados, por ejemplo, para el filtrado en conjuntos de datos que contengan ruido, concatenar diferentes nubes de puntos, segmentar partes relevantes, extraer puntos clave y para detectar primitivas geométricas como planos, cilindros, esferas o conos, elementos que conforman los elementos arquitectónicos típicos de las edificaciones históricas.

La biblioteca está en continuo desarrollo y en su mantenimiento están involucrados un gran número de ingenieros y científicos de diferentes organizaciones de todo el mundo. El proyecto está financiado por un gran consorcio de compañías entre las que se encuentran NVIDIA, Leica Geosystems, Intel, Google, Velodyne, Toyota, Honda y Spectrolab entre otras.

La biblioteca es multiplataforma y funciona bajo Linux, MacOS, Windows y Android o iOS. Para simplificar su desarrollo, está dividida en una serie de bibliotecas más pequeñas que pueden ser compiladas de forma independiente. Esta modularidad permite utilizar la biblioteca en sistemas embebidos que generalmente cuentan con restricciones tanto de

tamaño como computacionales. El grafo de dependencias de los diferentes módulos que componen la biblioteca ha sido representado en la Figura 4.1.

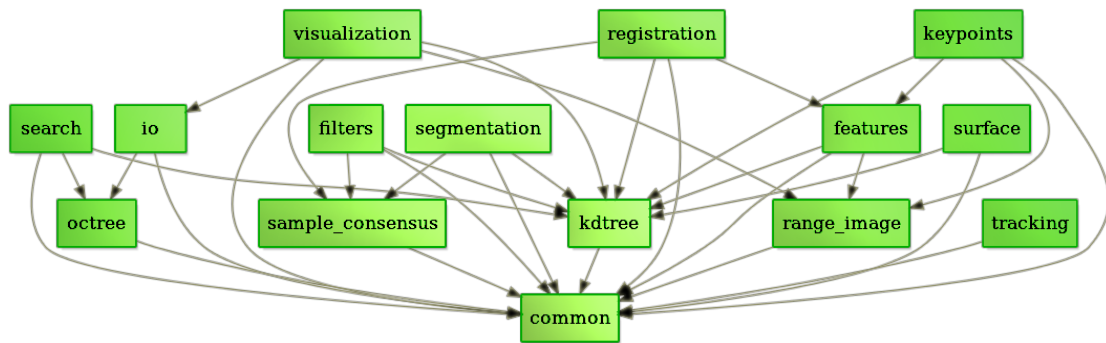


FIGURA 4.1: Estructura de la biblioteca PCL. (Fuente: web oficial de la biblioteca PCL[29]).

4.2. Algoritmos para la interpretación de nubes de puntos

La interpretación de la nube cuenta con una etapa inicial de importación de datos y cuatro etapas principales (ver Figura 4.2). Primero, la nube de puntos es preprocesada de tal forma que se elimina el ruido que contienen los datos y se disminuye el número de puntos del conjunto de partida. Una vez que el preprocesado ha sido completado se aplica un algoritmo de segmentación para dividir la nube en sus partes constituyentes para que puedan ser procesadas de forma independiente. Después, se aplica una técnica de clasificación por ajuste de modelos que determina si los segmentos pueden ser interpolados por planos o por cilindros. Finalmente, los diferentes segmentos son parametrizados y se establece la conectividad local de las diferentes secciones para que la geometría de la edificación pueda ser generada de forma adecuada. Mientras que las etapas de preprocesado y segmentación son ampliamente utilizadas en todo tipo de aplicaciones de tratamiento de nubes, las etapas de clasificación y análisis de dependencias representan un importante avance respecto a los desarrollos más modernos publicados de modelado automático.

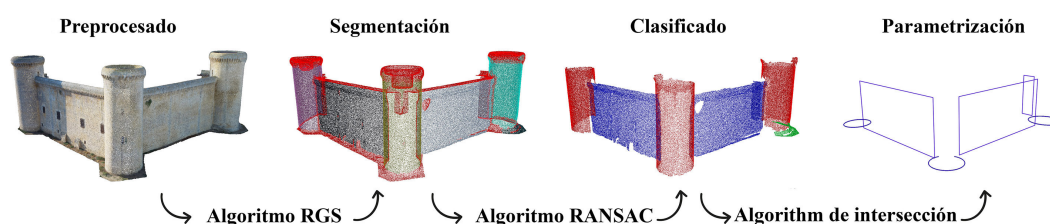


FIGURA 4.2: Esquema general del método propuesto. De izquierda a derecha. Preprocesado; Segmentación; Clasificación; Parametrización.

Los diferentes algoritmos han sido implementados utilizando el lenguaje de programación orientado a objetos C++. Se ha elegido este lenguaje por temas de eficiencia computacional y para asegurar una máxima compatibilidad con la biblioteca de tratamiento de nubes PCL, escrita en este mismo lenguaje. C++ es un lenguaje de programación de propósito general diseñado en 1979 por Bjarne Stroustrup que extiende el lenguaje C al incorporar nuevas funcionalidades como clases, sobrecarga de funciones, herencia o polimorfismo. C++ permite programar a bajo y alto nivel simultáneamente, lo que permite una alta optimización de los programas desarrollados.

4.2.1. Importación

La biblioteca PCL tiene predefinidos una gran cantidad de tipos de puntos básicos que pueden almacenar tanto información sobre el color tanto en el espacio de color RGB como en el espacio de color HSV. Se pueden definir nuevos tipos de puntos para almacenar características no soportadas de forma nativa como información termográfica. En la Tabla 4.1 se mencionan los principales tipos de puntos contemplados:

	X	Y	Z	R	G	B	I	N
PointXY	x	x	-	-	-	-	-	-
PointXYZ	x	x	x	-	-	-	-	-
PointXYZI	x	x	x	-	-	-	x	-
PointXYZRGB	x	x	x	x	x	x	-	-
PointXYZHSV	x	x	x	x	x	x	-	-
PointXYZRGBNormal	x	x	x	x	x	x	-	x
PointXYZINormal	x	x	x	-	-	-	x	x
PointXYZIRGBNormal	x	x	x	x	x	x	x	x

TABLA 4.1: Tipos de puntos soportados nativamente por la biblioteca PCL

Aunque existen herramientas que permiten la conversión entre formatos, para mejorar la usabilidad de la herramienta se ha desarrollado un lector que permite la importación de los formatos ptx, pts, ply, obj y pcd. Algunos de estos formatos son soportados directamente por la biblioteca PCL como el formato ply o el obj. Sin embargo, algunos de los formatos más utilizados en la industria, como el pts o el ptx, no lo son y ha sido necesario desarrollar funciones específicas para la importación de estos formatos. Una vez que el programa ha terminado de leer la nube se almacena en una estructura de tipo XYZRGB. Se debe destacar que el módulo de importación es capaz de detectar automáticamente el tipo de extensión del fichero cargado y seleccionar el sistema de lectura apropiado.

4.2.2. Preprocesado

La etapa de preprocesado consta de dos etapas, la etapa de filtrado y la etapa de remuestreo. Filtrar consiste en seleccionar un subconjunto de los datos de partida que contenga la mayor cantidad de información relevante sobre la escena escaneada con la menor pérdida de información posible. Durante el proceso se desechan los datos superfluos o erróneos que contienen las escenas digitalizadas debido a diferentes errores como el de medición. De esta forma se minimizan posibles problemas como imprecisiones en el modelo final o incluso fallos importantes en fases posteriores del algoritmo de procesamiento.

En la gran mayoría de los casos, los filtros estadísticos comparan características de cada uno de los puntos de la nube con el valor medio de esa característica dentro de la nube de puntos y elimina aquellos que no cumplen un cierto criterio. El filtro seleccionado utiliza como criterio la distancia media que separa cada uno de los puntos de la nube $\{P\}$ de sus vecinos. Para cada punto, se calcula la distancia media a sus vecinos obteniendo una distancia media global a nivel de nube a partir de las medias asociadas a cada punto. Asumiendo una distribución Gaussiana, todos los puntos cuya distancia media se encuentre fuera del intervalo definido por la media global y la desviación normal global son considerados ruido y eliminados del conjunto de puntos de la nube.

Siendo:

- Función para determinar los vecinos del punto: $\Omega(x)$

- Distancia en valor absoluto entre dos puntos: $\Psi(P_1, P_2)$
- Desviación estándar: $\sigma(\{x\})$
- Media: $\overline{\{x\}}$

Algorithm 1 Filtrado estadístico de nubes de puntos

```

1: function ALGORITMO( $\{P\}$ )
2:    $D \leftarrow \emptyset$  ▷ Distancias medias
3:   for  $i = 1$  to  $size(\{P\})$  do
4:      $\{V\} \leftarrow \Omega(P_i)$  ▷ Vecinos del punto i-ésimo
5:      $d = 0$ 
6:     for  $j = 1$  to  $size(\{V\})$  do
7:        $\{D\} \leftarrow \{D\} \cup \Psi(P_i, V_j)$  ▷ Distancia media del punto
8:     end for
9:   end for
10:  for  $i = 1$  to  $size(\{D\})$  do
11:    if  $D_i > \overline{\{D\{P_i\}}} + \sigma(\{D\{P_i\}\})$  then
12:       $\{P\} \leftarrow \{P\} \setminus P_i$  ▷ Punto outlier
13:    end if
14:  end for
15: end function

```

Por otra parte, el remuestreo consiste en alterar el número de puntos de la nube de partida sin alterar las dimensiones geométricas o resolución del conjunto de datos de partida añadiendo o eliminando datos de las escenas digitalizadas. El remuestreo, aplicado a nubes de puntos, generalmente se emplea para reducir el número de puntos buscando aumentar la velocidad de procesamiento de etapas posteriores computacionalmente más pesadas. El algoritmo genera un árbol octal a partir del conjunto de datos y reemplaza los puntos que pertenezcan al mismo nodo por su centroide. Esta etapa de procesamiento, aunque no es necesaria, reduce enormemente el tiempo de computación global si se ajusta adecuadamente la compresión de la nube.

4.2.3. Segmentación

El análisis de interdependencia entre primitivas es un problema topológico clave que debe ser resuelto para la correcta interpretación de la nube. Analizar la nube de puntos de una sola vez es una tarea prácticamente imposible que puede llevar a tiempos de procesamiento extremadamente largos, resultados poco adecuados y poco precisos o incluso a fallos en la mayor parte de los casos.

La segmentación de la nube es una etapa fundamental de la metodología propuesta y consiste en clasificar cada uno de los puntos de la nube en función de su pertenencia a uno de los posibles segmentos que forman parte del conjunto. De esta forma, puntos que sean parte del mismo segmento compartirán ciertas características.

Para realizar una correcta segmentación es necesario determinar como paso previo las normales superficiales de la nube. En una superficie geométrica es generalmente sencillo inferir la dirección de la normal asociada a un punto concreto ya que basta con determinar la dirección perpendicular a la superficie en ese punto. Sin embargo, como las nubes de puntos son conjuntos de puntos dispersos, no es trivial determinar la dirección normal a la "superficie" generada por los puntos. Para calcular las normales existen dos posibilidades:

- Obtener la superficie subyacente a partir del conjunto de puntos utilizando técnicas de mallado y calcular las normales de la malla.
- Utilizar aproximaciones para inferir las normales directamente a partir de la nube.

Para el desarrollo de la aplicación se ha optado por emplear una aproximación que determina la normal del punto aproximando dicha normal por la del plano tangente a la superficie generada por los puntos cercanos, que en la práctica consiste en determinar la ecuación de este plano realizando una optimización por mínimos cuadrados.

El cálculo de las normales superficiales matemáticamente se reduce, por tanto, al análisis de los autovectores y autovalores (análisis de componentes principales) de la matriz de covarianza creada empleando los vecinos más cercanos del punto estudiado. Más específicamente, para cada punto \mathbf{p}_i , la matriz de covarianza \mathcal{C} se construye según la expresión 4.1.

$$\mathcal{C} = \frac{1}{k} \sum_{i=1}^k (\mathbf{p}_i - \bar{\mathbf{p}}) \cdot (\mathbf{p}_i - \bar{\mathbf{p}})^T, \mathcal{C} \cdot \vec{\mathbf{v}}_j = \lambda_j \cdot \vec{\mathbf{v}}_j, j \in \{0, 1, 2\} \quad (4.1)$$

Donde k es el número de puntos vecinos considerados en el vecindario de \mathbf{p}_i , $\bar{\mathbf{p}}$ representa el centroide 3D de los vecinos más cercanos, λ_j es el j -ésimo autovalor de la matriz de covarianza, y $\vec{\mathbf{v}}_j$ el j -ésimo autovector.

Una vez determinadas las normales se aplica el algoritmo de Segmentación de Regiones Crecientes (RGS) que se utiliza para relacionar los puntos que están suficientemente

cerca en términos de orientación de las normales superficiales. De este modo, el algoritmo determina el conjunto de regiones que compone la nube, siendo cada región un grupo de puntos que son considerados parte de la misma superficie.

Inicialmente se ordenan los puntos por los valores de sus curvaturas c . Este paso es necesario porque la región comienza a crecer a partir del punto con valor mínimo de curvatura. El punto con la mínima curvatura se encuentra en la zona plana de la superficie que genera la nube y al seguir esta estrategia se minimiza el número total de segmentos. Una vez que se ha seleccionado el punto inicial de crecimiento el proceso seguido es el siguiente:

1. El punto seleccionado es añadido al conjunto de puntos semilla.
2. Para cada punto semilla el algoritmo busca los puntos vecinos.
 - a) Se comprueba el ángulo que forma la normal de cada vecino y la normal de la semilla. Si el ángulo es menor que un cierto valor entonces el punto es añadido a la región.
 - b) Tras esto se comprueba el valor de la curvatura de cada vecino. Si la curvatura es menor que un cierto valor entonces el punto es añadido a las semillas.
 - c) La semilla actual es eliminada del conjunto de semillas.

Si el conjunto de las semillas se queda vacío entonces el algoritmo ha finalizado el crecimiento de la región y el proceso se repite desde el principio para generar la siguiente región. El Algoritmo 2 recoge el pseudo-código del algoritmo RGS siendo:

- Nube de puntos: $\{P\}$
- Normales: $\{N\}$
- Función de búsqueda de vecinos: $\Omega(x)$
- Función de búsqueda de punto con mínima curvatura: Ψ
- Umbral de curvatura: c_{th}
- Umbral de ángulos: θ_{th}

Algorithm 2 Segmentación de nubes de puntos con algoritmo RGS

```

1: function ALGORITMO( $\{P\}$ ,  $\{N\}$ ,  $\{c\}$ ,  $c_{th}$ ,  $\theta_{th}$ )
2:    $R \leftarrow \emptyset$  ▷ Regiones
3:   while  $A \neq \emptyset$  do
4:      $\{Rc\} \leftarrow \emptyset$  ▷ Región actual
5:      $\{S\} \leftarrow \emptyset$  ▷ Semillas
6:      $P_{min} \leftarrow \Psi(\{P\})$ 
7:      $\{S\} \leftarrow \{S\} \cup P_{min}$ 
8:      $\{Rc\} \leftarrow \{Rc\} \cup P_{min}$ 
9:      $\{P\} \leftarrow \{P\} \setminus P_{min}$ 
10:    for  $i = 1$  to  $size(\{S\})$  do
11:       $\{B\} \leftarrow \Omega(S_i)$  ▷ Vecinos cercanos a la semilla
12:      for  $j = 1$  to  $size(\{B\})$  do
13:        if  $B_j \in \{P\} \wedge \cos^{-1}(|(N\{S_i\}, N\{S_j\})|) < \theta_{th}$  then
14:           $\{Rc\} \leftarrow \{Rc\} \cup B_j$ 
15:           $\{P\} \leftarrow \{P\} \setminus B_j$ 
16:          if  $c\{B_j\} < c_{th}$  then
17:             $\{S\} \leftarrow \{S\} \cup B_j$ 
18:          end if
19:        end if
20:      end for
21:    end for
22:     $\{R\} \leftarrow \{R\} \cup \{Rc\}$ 
23:  end while
24:  return  $\{R\}$ 
25: end function

```

Los algoritmos aplicados hasta el momento son ampliamente utilizados para el preprocesado de grandes conjuntos de información y no es necesario ajustar parámetros con la excepción del algoritmo de remuestreo, en el que se ajusta el tamaño del nodo del árbol generado. Por otra parte, la etapa de segmentación emplea unos pocos parámetros de configuración: un valor de límite para la desviación de las normales vecinas, un valor límite para la curvatura y, opcionalmente, un límite para el número mínimo de puntos que debe tener una región. Los valores óptimos de desviación y curvatura fueron determinados experimentalmente mientras que el tamaño mínimo del segmento ha sido establecido proporcionalmente en base al número de puntos de la nube de partida.

4.2.4. Clasificación

El objetivo de la etapa de clasificación es asignar una categoría a cada nube de puntos segmentada a partir de la nube de partida en función de su forma externa. Cabe destacar que, debido a la gran complejidad del problema, el algoritmo hace algunas suposiciones

sobre la forma de las nubes de puntos analizadas. Por lo tanto, el algoritmo se centra en el análisis de muros y torres, aunque se podrían agregar características arquitectónicas más avanzadas, como cúpulas o bóvedas, tal y como se analiza en las conclusiones. Los diferentes segmentos se etiquetan de acuerdo con el modelo matemático que mejor se ajusta a los puntos de la región, a saber: 1) planos; 2) cilindros; 3) e indefinido. La etapa de clasificación emplea el algoritmo estadístico RANSAC (RANdom SAmple Consensus; Fischler, 1981), que es capaz de estimar los parámetros de un modelo ajustado a través de un proceso iterativo, asumiendo que los datos de inicio se componen de inliers y outliers. Los inliers (puntos más cercanos al modelo que una cierta distancia de umbral especificada por el usuario) se utilizan para estimar el modelo, mientras que los outliers son descartados durante el proceso de estimación del modelo.

El algoritmo RANSAC alcanza su objetivo seleccionando de forma iterativa a un conjunto aleatorio de los datos de partida. Estos datos son hipotéticos puntos del modelo y la hipótesis realizada es verificada de la siguiente forma:

- Se ajusta un modelo matemático a los hipotéticos puntos que pertenecen a éste.
- El resto de puntos son testados frente al modelo ajustado y si un punto se ajusta correctamente al modelo se le considera también un hipotético punto del conjunto.
- El modelo estimado es razonablemente bueno si han sido clasificados suficientes puntos como puntos hipotéticos del modelo.
- Se re-estiman los parámetros del modelo empleando todos los puntos hipotéticos porque sólo se habían estimado los parámetros a partir del conjunto inicial de puntos.
- Finalmente, el modelo es evaluado estimando el error de los puntos hipotéticos del modelo.

Este procedimiento se repite un número fijo de veces, cada una de ellas produciendo bien un modelo que es rechazado porque muy pocos puntos son considerados como pertenecientes al modelo o bien un modelo refinado junto a una medida del error de ajuste. El modelo refinado se conserva para la iteración siguiente si su error es menor que el del modelo anterior. El pseudo-código del RANSAC ha sido descrito en el algoritmo 3 siendo:

- Nube de puntos: $\{P\}$
- Función que calcula los parámetros del modelo: $\Omega(x)$
- Función que calcula el grado de pertenencia de un punto a un modelo: $\phi(x)$
- Mínimo número de puntos ajustados: n
- Máximo número de iteraciones: k
- Umbral de pertenencia del punto al modelo: t
- Número de puntos cercanos al modelo que aseguran que el modelo es válido: d

Algorithm 3 Cálculo de modelos de ajuste utilizando algoritmo RANSAC

```

1: function RANSAC( $\{P\}, n, k, d$ )
2:    $indice = 0$ 
3:    $\Pi \leftarrow \emptyset$ 
4:    $error = \infty$ 
5:   while  $indice < k$  do
6:      $\{L\} \subseteq \{P\}$  ▷ Posibles inliers aleatorios de P
7:      $\{L^c\} = \{L\} \setminus \{P\}$  ▷ Complementario de L
8:      $\Pi^* \leftarrow \Omega(\{L\})$  ▷ Parámetros del posible modelo
9:      $\{L^*\} \leftarrow \emptyset$  ▷ Otros posibles inliers
10:    for  $i = 1$  to  $size(\{L^c\})$  do
11:      if  $\phi(\Pi^*, \{L_i^c\}) < t$  then
12:         $L^* \leftarrow L^* \cup L_i^c$  ▷ El modelo ajusta el punto
13:      end if
14:    end for
15:    if  $size(\{L^*\}) > d$  then ▷ Posible buen modelo
16:       $\Pi^* \leftarrow \Omega(\{L\} \cup \{L^*\})$ 
17:       $error^* = f_e(\Pi^*, \{L\} \cup \{L^*\})$ 
18:      if  $error^* < error$  then ▷ Mejora el modelo anterior
19:         $\Pi \leftarrow \Pi^*$ 
20:         $error = error^*$ 
21:      end if
22:    end if
23:     $indice = indice + 1$ 
24:  end while
25:  return  $\Pi, error$ 
26: end function

```

Una ventaja del algoritmo RANSAC es su capacidad de estimar de forma robusta los parámetros del modelo incluso cuando hay una cantidad significativa de ruido en los datos de partida. Sin embargo, el algoritmo no tiene un límite superior para el tiempo que tarda en calcular los parámetros. Además, el número de iteraciones debe ser introducido

manualmente y la solución obtenida puede no ser óptima o que incluso no sea una que ajuste los datos de forma correcta si el modelo elegido no es adecuado.

También es importante destacar que el algoritmo RANSAC puede estimar sólo un modelo a partir de un conjunto de datos. Cuando dos modelos coexisten en el mismo conjunto de datos el algoritmo RANSAC fallará al buscar el segundo, motivo por el que la nube fue segmentada anteriormente en conjuntos más sencillos.

Por otra parte, que un modelo ajuste correctamente un alto porcentaje de puntos no implica que dicho modelo sea el más idóneo. Por ejemplo, una nube puede ser aproximada por un plano con un alto grado de ajuste, pero también podría ser ajustada usando como modelo matemático un cilindro de radio suficientemente elevado. Para evitar este importante problema se ha implementado un algoritmo de clasificación, que selecciona el mejor modelo de ajuste de entre los mencionados. Los parámetros tanto del plano como del cilindro se estiman para cada conjunto de datos y el modelo que explica la mayor cantidad de puntos se selecciona si ajusta un porcentaje suficientemente elevado de inliers, un parámetro especificado manualmente por el usuario. De lo contrario, el segmento se clasifica como indefinido. Una vez que se valida el modelo, los inliers son proyectados sobre la superficie calculada y se guardan para un uso posterior mientras que los outliers son descartados.

4.2.5. Parametrización

El algoritmo RANSAC estima los parámetros de los planos y los cilindros mediante la ecuación general que representa matemáticamente estas dos superficies. En los planos se determinan los parámetros A, B, C y D de la ecuación implícita $\Pi \equiv A \cdot x + B \cdot y + C \cdot z + D = 0$ mientras que para cilindros se obtiene la ecuación del eje en su forma punto-pendiente $f(x, y, z) = x_0 + \lambda \cdot \vec{v}$ y el valor del radio R . Sin embargo, las ecuaciones definen superficies infinitas en las que es necesario establecer límites geométricos en una etapa posterior de parametrizado.

4.2.5.1. Muros

Se hacen realizando dos hipótesis sobre las paredes: 1) pueden ser modeladas a través de planos; 2) tienen forma rectangular. Esta simplificación facilita y robustece etapas

posteriores de procesamiento y aunque la simplificación pueda parecer excesiva no hay gran pérdida de información ya que el enfoque seguido intenta recrear el modelo 3D de la construcción original, no el modelo de la construcción según se encuentra en el momento de la captura de la información. Para parametrizar los muros se aplica un algoritmo que determina los cuatro vértices del rectángulo.

La primera etapa consiste en ampliar las dimensiones de la nube $\{P\}$ correspondiente a uno de los muros. Durante la segmentación de la nube, los puntos limítrofes de los segmentos son descartados debido a las irregularidades de las normales superficiales, reduciendo el área total de la nube en comparación con la superficie de la pared teórica.

La ampliación de la nube del plano se consigue proyectando puntos en las aristas teóricas definidas por las intersecciones con el resto de muros. Las rectas R que definen las aristas se obtienen como la intersección del plano Π del muro estudiado con el resto de los planos detectados $\{\Lambda\}$. El Algoritmo 4 de intersección de planos proyecta tanto los puntos de la nube P asociada al plano Π como los puntos de las nubes $\{S\}$ asociadas a los planos $\{\Lambda\}$ sobre la recta R .

Si tanto los puntos proyectados de $\{P\}$ como los proyectados de $\{S\}$ están a una distancia menor que una tolerancia θ_p de los puntos de $\{P\}$ se considera que R es una recta que tiene en común los dos planos y los puntos proyectados de $\{P\}$ son incorporados a la nube $\{P\}$ maximizando de esta forma sus dimensiones y consiguiendo que se reflejen más fielmente las dimensiones reales de los muros. Matemáticamente, que dos conjuntos estén cerca ha sido interpretado como que al menos un punto del conjunto $\{P\}$ está a una distancia de uno de los puntos proyectados de $\{P\}$ menor que una cierta tolerancia. El pseudo-código 4 resume el funcionamiento del algoritmo descrito siendo $\Psi(\{X\}, \{Y\})$ la función de cálculo de distancia entre nubes.

Para ampliar las dimensiones de los planos adyacentes a cilindros se aplica un algoritmo similar (Algoritmo 5). Los puntos del plano $\{P\}$ son proyectados sobre los modelos matemáticos de los cilindros $\{C\}$ obteniendo el conjunto de puntos $\{P_p\}$. El conjunto $\{P_p\}$ vuelve a ser proyectado sobre el plano Π asociado a la nube $\{P\}$ para garantizar la coplanaridad de $\{P_p\}$ y $\{P\}$. Si los puntos proyectados están a una distancia menor que una cierta tolerancia θ_c de los puntos del $\{P\}$ se considera que están en contacto directo con el cilindro y el nuevo conjunto de puntos se añade a $\{P\}$.

Algorithm 4 Ampliación de nubes por cálculo de intersecciones plano-plano

```

1: function AMPLIAR( $P, \Pi, \{S\}, \{\Lambda\}, \theta_p$ )
2:   for  $i \leftarrow 1$  to  $size(\{\Lambda\})$  do
3:      $R \leftarrow \Pi \cap \Lambda_i$ 
4:      $\{P_p\} \leftarrow proj_R\{P\}$ 
5:      $\{S_p\} \leftarrow proj_R\{S_i\}$ 
6:     if  $\Psi(\{P_p\}, \{P\}) < \theta_p \wedge \Psi(\{S_p\}, \{P\}) < \theta_p$  then
7:        $\{P\} \leftarrow \{P\} \cup \{P_p\}$ 
8:     end if
9:   end for
10:  return  $\{P\}$ 
11: end function

```

Algorithm 5 Ampliación de nubes por cálculo de intersecciones plano-cilindro

```

1: function AMPLIAR( $\{P\}, \{C\}, \Pi$ )
2:   for  $i = 1$  to  $size(\{C\})$  do
3:      $\{P_p\} \leftarrow proj_{C_i}\{P\}$ 
4:      $\{P_{pp}\} \leftarrow proj_{\Pi}\{P_p\}$ 
5:     if  $\Psi(\{P_{pp}\}, \{P\}) < \theta_c$  then
6:        $\{P\} \leftarrow \{P\} \cup \{P_{pp}\}$ 
7:     end if
8:   end for
9:   return  $\{P\}$ 
10: end function

```

En la segunda etapa se determinan los 4 vértices V_i del plano a partir de los puntos que componen la nube $\{P\}$ ampliada. El algoritmo recorre los puntos de $\{P\}$ y almacena los máximos y mínimos valores registrados de las coordenadas en los puntos auxiliares $V_{max}(x_{max}, y_{max}, z_{max})$ y $V_{min}(x_{min}, y_{min}, z_{min})$.

A partir de V_{max} y V_{min} se obtienen 4 vértices teóricos $\{V_t\}$ para los dos vértices inferiores del plano, es decir, los vértices con menor cota z . El conjunto $\{V_t\}$ se obtiene combinando los valores de las coordenadas (x, y) de V_{max} y V_{min} manteniendo constante el valor de $z = V_{min,x}$ (Tabla 4.2).

Vértice	x	y	z
V_{t1}	x_{min}	y_{min}	z_{min}
V_{t2}	x_{min}	y_{max}	z_{min}
V_{t3}	x_{max}	y_{min}	z_{min}
V_{t4}	x_{max}	y_{max}	z_{min}

TABLA 4.2: Vértices teóricos

En la Tabla 4.3 se adjunta un ejemplo con valores obtenidos al aplicar el algoritmo de discriminación de vértices teóricos siendo $P_{min} = \{1.2, 2.4, 0.1\}$ y $P_{max} = \{5.1, 7.3, 4.1\}$

Vértices	x	y	z	d	Apto
V_{t1}	1.2	2.4	0.1	0.3	✓
V_{t2}	1.2	7.3	0.1	0.1	✓
V_{t3}	5.1	2.4	0.1	12.1	✗
V_{t4}	5.1	7.3	0.1	15.3	✗

TABLA 4.3: Ejemplo de valores típicos de distancias

Una vez generados los vértices se mide la distancia d de cada uno de ellos al plano matemático empleando la fórmula 4.2 y se descartan los dos más alejados. Los dos vértices teóricos más cercanos al plano se almacenan como vértices temporalmente.

Los vértices superiores del plano se obtienen sustituyendo el valor de la cota z de los vértices almacenados por el valor de $V_{max,z}$. En la Tabla 4.4 se muestran los valores que tendrían las coordenadas de los vértices del ejemplo anterior.

Vértices	x	y	z
V_{p1}	1.2	2.4	0.1
V_{p2}	1.2	7.3	0.1
V_{p3}	1.2	2.4	4.1
V_{p4}	1.2	7.3	4.1

TABLA 4.4: Conjunto típico de vértices para un muro rectangular

$$d = \frac{|A \cdot V_x + B \cdot V_y + C \cdot V_z + D|}{\sqrt{A^2 + B^2 + C^2}} \quad (4.2)$$

Estos cuatro vértices $\{V_p\}$ se proyectan sobre el plano y se almacenan temporalmente quedando a la espera de ser refinados en una etapa posterior de procesamiento.

La tercera y última etapa consiste en refinar la posición de los vértices $\{V_p\}$ empleando las ecuaciones de las rectas que definen las aristas de los planos. Calculando la intersección entre las diferentes rectas es posible sustituir vértices de $\{V_p\}$ por puntos definidos por estas intersecciones que tienen una mayor precisión.

El algoritmo de sustitución (Algoritmo 6) determina las intersecciones V_i que existen entre las rectas R que definen las aristas del plano. Las coordenadas de los vértices V_i son comparadas con las de los vértices de V_p . Si uno de los vértices de V_i se encuentra a una distancia menor que θ de uno de los vértices de $\{V_p\}$ se sustituye el vértice de $\{V_p\}$ por el de V_i . El Algoritmo 6 recoge el pseudo-código de la función de refinamiento de

vértices siendo $\Psi(P_1, P_2)$ la distancia euclídea entre dos puntos y $\Omega(\{P\})$ la función de reajuste de coordenadas.

Si se produce la sustitución de un vértice, las coordenadas del resto de vértices del $\{V_p\}$ son modificadas en función de las coordenadas del nuevo vértice para mantener las proporciones del rectángulo.

Algorithm 6 Refinamiento de vértices con intersecciones teóricas

```

1: function AMPLIAR( $V_i, V_p[ ]$ )
2:   for  $i = 1$  to  $size(\{V_p\})$  do
3:     if  $\Psi(V_i, V_p[i]) < \theta$  then
4:        $V_p[i] \leftarrow V_i$ 
5:        $\{V_p\} \leftarrow \Omega(\{V_p\})$ 
6:     end if
7:   end for
8: end function

```

El diagrama de flujo del algoritmo propuesto ha sido representado en la Figura 4.2.5.1.

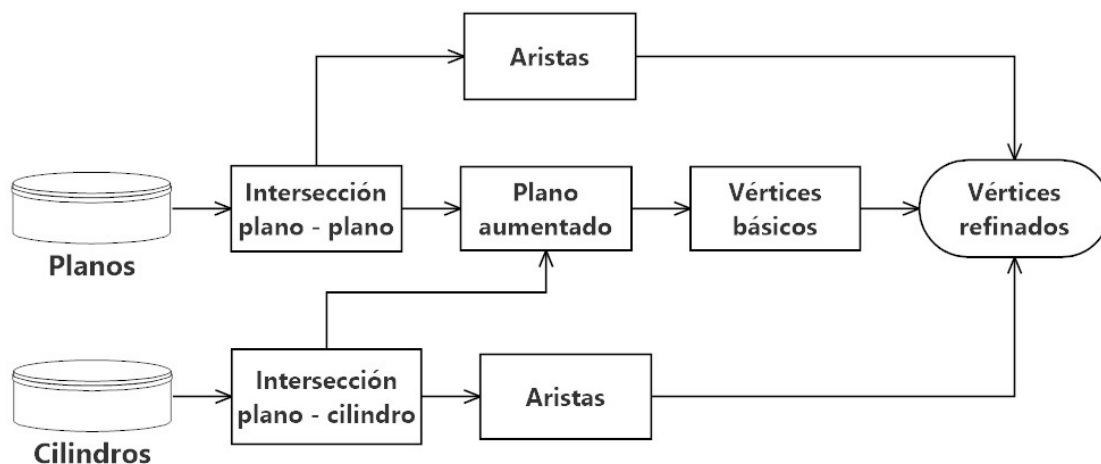


FIGURA 4.3: Diagrama de flujo de la metodología propuesta para el cálculo de los vértices de los muros.

4.2.5.2. Torres

Para parametrizar las torres se ha partido de la premisa de que pueden ser modeladas como cilindros. El algoritmo RANSAC parametriza los cilindros con la ecuación punto-pendiente $(R(x, y, z) = p_0(x_0, y_0, z_0) + \vec{v} \cdot \lambda)$ del eje y la longitud del radio r . Si \vec{v} se desvía más de un cierto ángulo respecto al eje OZ se descarta el cilindro como posible candidato a torre. Si por el contrario la dirección del vector se da por válida se impone su paralelismo respecto a OZ .

Para determinar los límites del cilindro se han parametrizado las bases de la torre. Se han calculado las coordenadas del centro de las mismas ya que, junto con los parámetros del modelo ya conocidos, es posible definir completamente los límites del cilindro. Para parametrizar los centros se determina el valor de la altura mínima (z_{min}) y máxima (z_{max}) que aparece en la nube del cilindro. Con z_{min} y z_{max} se definen dos planos Π_1 y Π_2 paralelos al plano XY que pasan cada uno de ellos por uno de los puntos. Las coordenadas de las bases (V_1, V_2) se obtienen al proyectar el punto P_0 sobre estos planos. Finalmente la altura del cilindro h se obtiene como la distancia entre z_{min} y z_{max} .

El pseudo-código de la función de parametrización de cilindro es descrita en el Algoritmo 7 siendo:

- Nube de puntos del cilindro: $\{P\}$
- Proyección de punto sobre modelo: $\Omega(\Pi, P_i)$
- Umbral máximo de desviación del vector director: θ

Algorithm 7 Parametrización de torres

```

1: function PARAMETRIZAR( $\{P\}, P_0, R, \theta$ )
2:   if  $v_z < \theta$  then                                     ▷ No es vertical
3:     return
4:   else
5:      $v_z = 1.0$ 
6:      $\vec{v} \leftarrow \vec{v} / \|\vec{v}\|$ 
7:   end if
8:    $z_{min} = \infty, z_{max} = -\infty$ 
9:   for  $i = 0$  to  $size(\{P\})$  do
10:    if  $P_{iz} > z_{max}$  then
11:       $z_{max} = P_{iz}$ 
12:    end if
13:    if  $P_{iz} < z_{min}$  then
14:       $z_{min} = P_{iz}$ 
15:    end if
16:  end for
17:   $\Pi_1(x, y, z) \equiv C_1 \cdot z + D_1 = 0$                        ▷ Plano perpendicular a eje  $OZ$ 
18:   $\Pi_2(x, y, z) \equiv C_2 \cdot z + D_2 = 0$ 
19:   $\Pi_1(0, 0, z_{min}) \equiv D_1 = -C_1 \cdot z_{min} \Rightarrow \Pi_1 \equiv z = z_{min}$ 
20:   $\Pi_2(0, 0, z_{max}) \equiv D_2 = -C_2 \cdot z_{max} \Rightarrow \Pi_2 \equiv z = z_{max}$ 
21:   $V_1 \leftarrow \Omega(\Pi_1, P_0)$ 
22:   $V_2 \leftarrow \Omega(\Pi_2, P_0)$ 
23:  return  $V_1, V_2$ 
24: end function

```

4.3. Integración con Revit

Los algoritmos desarrollados han sido integrados dentro de una biblioteca dinámica (DLL) que puede ser utilizada por otras aplicaciones. De esta forma, una gran cantidad de aplicaciones, entre las que se encuentra Revit, pueden aprovechar los resultados obtenidos tras el procesamiento de las nubes. En el caso de Revit, su sistema de plug-ins permite integrar todas las funcionalidades desarrolladas de forma sencilla ya que este software pone a disposición de los desarrolladores un extenso kit de desarrollo de software (SDK) que permite acceder a la vasta biblioteca de funcionalidades del software.

Se ha desarrollado un plug-in para Revit que ofrece una interfaz sencilla e intuitiva con la que el usuario es capaz de seleccionar la nube de puntos a procesar y de modificar los parámetros de los algoritmos desarrollados dentro de la propia aplicación. Además, el plug-in genera al finalizar el procesamiento de la nube el modelo sólido de los elementos detectados con los que se puede interactuar directamente en Revit.

4.3.1. Interfaz

Antes de utilizar la herramienta se debe crear un nuevo proyecto dentro de Revit. Una vez que se ha creado el proyecto aparece un nuevo icono en la cinta de opciones disponibles bajo el menú *Complementos* (Figura 4.4) que permita al usuario inicializar el plug-in desarrollado y desde donde se pueden modificar los diferentes parámetros del algoritmo de procesamiento (Figura 4.5).

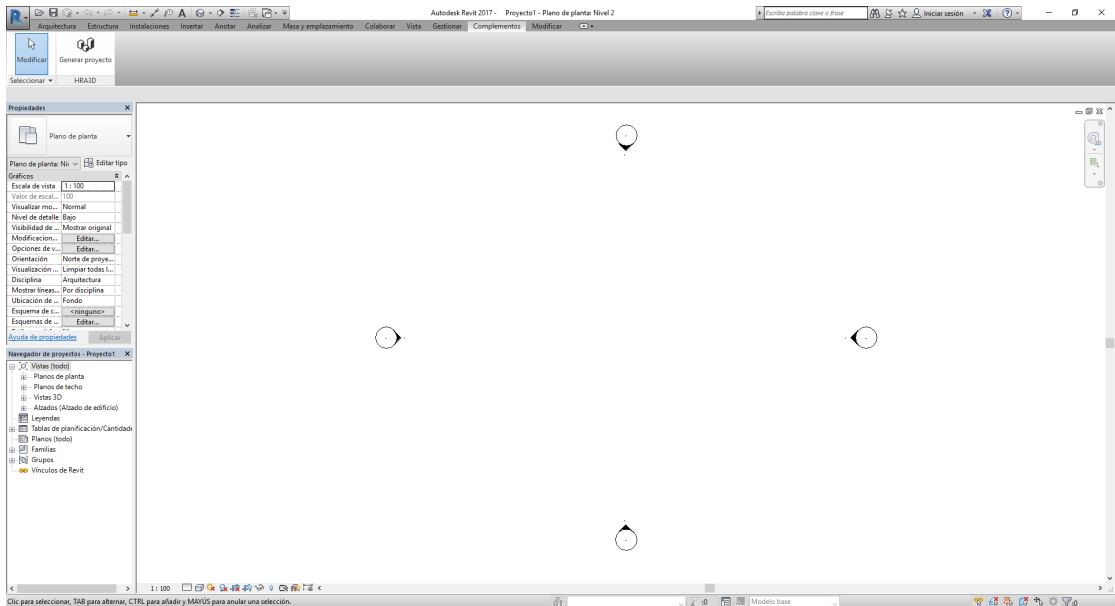


FIGURA 4.4: Cinta de selección de plug-ins de Revit (parte superior izquierda).

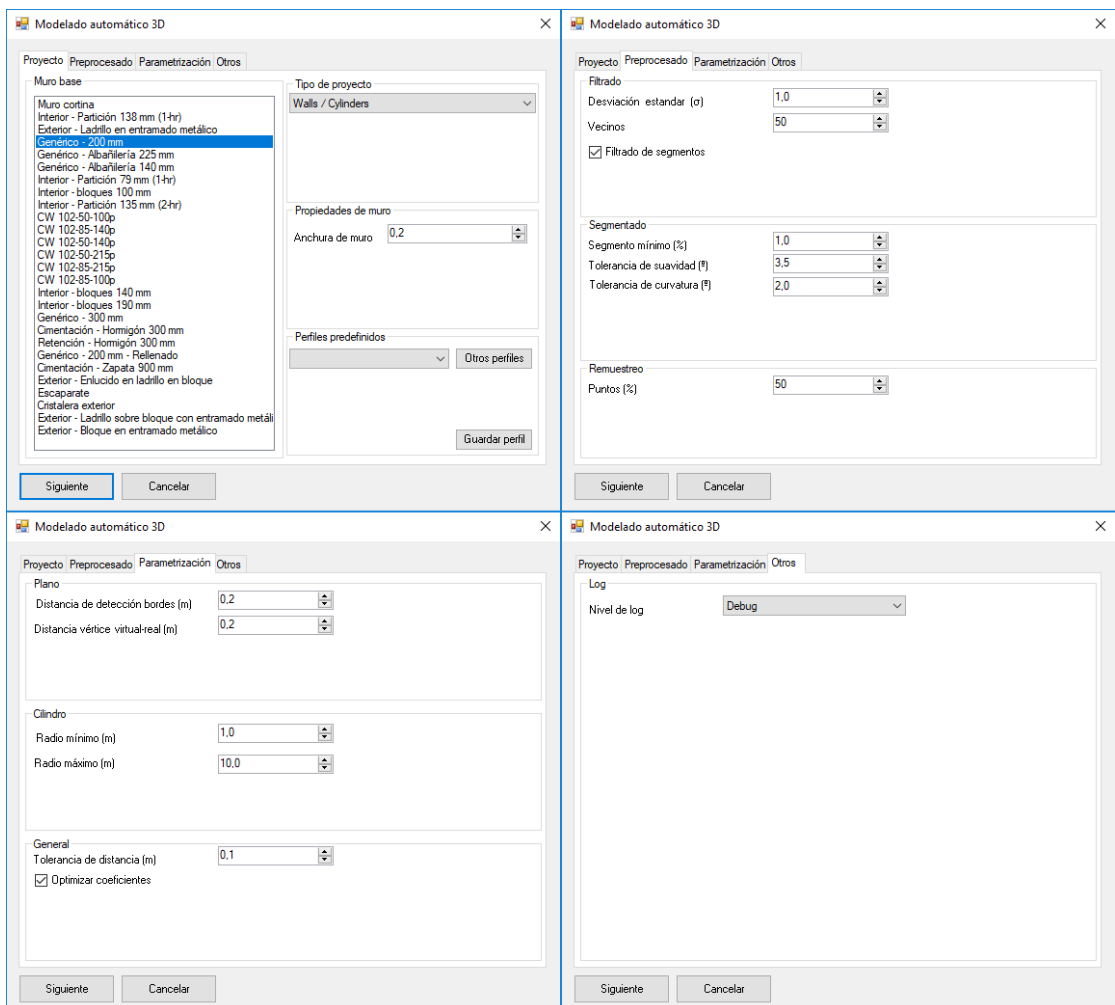


FIGURA 4.5: Pantallas de configuración del plug-in.

Al presionar el botón se despliega un formulario (Figura 4.6) donde se pueden modificar los parámetros de configuración descritos en la Tabla 4.5. Estos parámetros cubren todo el flujo de trabajo del procesamiento automático de las nubes de puntos.

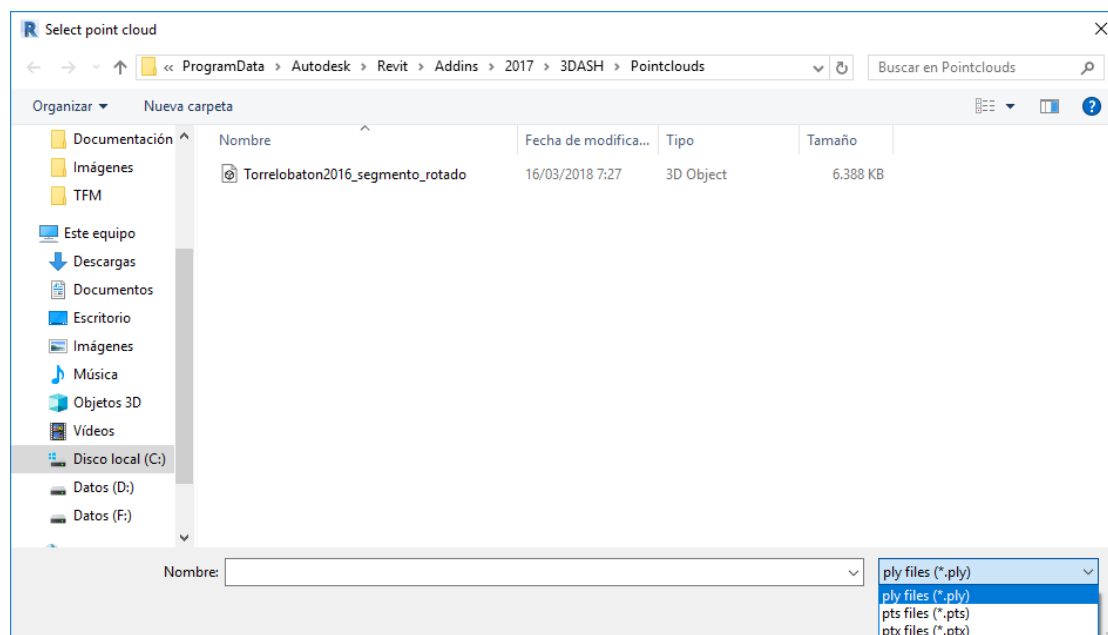


FIGURA 4.6: Formulario de selección de nube de puntos

Una vez que se han establecido los parámetros se puede presionar el botón **Siguiente**. El programa muestra un nuevo formulario donde se puede seleccionar la nube de puntos.

Proyecto	Muro base Tipo de proyecto Propiedades de muro Perfiles predefinidos	Tipo de muro de las primitivas generadas Tipo de primitivas detectadas Anchura del muro generado Cargar y guardar perfiles de parámetros
Preprocesamiento	Filtrado Segmentación Remuestreado	Filtrado de outliers de la nube de puntos Modificar parámetros del algoritmo de segmentación Disminuir el porcentaje de puntos de la nube
Parametrización	General Planos Cilindros	Tolerancia del clasificador Tolerancia de detección de aristas comunes Tolerancia de detección de aristas comunes
Miscelánea	Log	Modificar la cantidad de información registrada en el archivo log

TABLA 4.5: Parámetros de configuración del plug-in para Revit

Una vez que se selecciona la nube de puntos y se presiona el botón de continuar el algoritmo procesa el conjunto de puntos seleccionado y genera elementos paramétricos que utilizan las propiedades de la biblioteca cargada cuando se creó el nuevo proyecto. Los elementos generados pueden ser exportados fácilmente al estándar IFC.

Capítulo 5

Resultados

En este capítulo se describen las pruebas realizadas con las que se han probado diferentes aspectos técnicos de la metodología desarrollada entre los que se puede destacar el tiempo de procesamiento o la fiabilidad del modelo final entre otros. La metodología ha sido testada utilizando dos nubes de puntos diferentes: una nube sintética y un fragmento representativo de la nube del Castillo de Torrelobatón.

5.1. Ensayos realizados

Se han probado los diferentes algoritmos desarrollados sobre una nube de puntos sintética generada a partir de fragmentos de la nube de puntos del Castillo de Torrelobatón con la que se pretende probar el desempeño general del sistema y la detección de encuentros tanto entre planos como entre planos y cilindros. Adicionalmente, el sistema ha sido probado con un fragmento real de la nube de puntos del castillo. Las pruebas se han llevado a cabo en un equipo de sobremesa que cuenta con Windows 10, una CPU Intel(R) Core i5 3470@3.2GHz y 16GB de memoria RAM. Los tiempos de procesamiento así como la selección de parámetros para ambas nubes han sido reflejados en la Tabla [5.1](#).

Etapa	Sub-etapa	Parámetro	Real		Sintético		Nota
			Tiempo(s)	Valor	Tiempo(s)	Valor	
Lectura de datos	Preprocesado		1.45		8.03		
		Filtrado	0.08	50	0.75	50	Asumido
Segmentado	Normales superficiales Segmentado	Tamaño de voxel	0.04	0.60	0.18	0.4	Especificado por usuario
		Puntos vecinos	0.58	50	2.06	50	Asumido
		Límite de suavidad	0.50	0.04	1.67	0.02	Especificado por usuario
		Límite de curvatura		0.05		0.05	Especificado por usuario
Clasificado		Segmento mínimo %		1		1	Especificado por usuario
		Límite de inliers (plano)	0.14	0.30	0.39	0.25	Especificado por usuario
		Límite de inliers (cilindro)		0.30		0.25	Especificado por usuario
		Porcentaje mínimo de inliers		75		75	Especificado por usuario
Parametrizado		Distancia límite plano-plano	0.05	2	1.25	2.5	Especificado por usuario
		Distancia límite cilindro-plano		2		1.5	Especificado por usuario
			2.84		14.32		Tiempo total

TABLA 5.1: Selección de parámetros y tiempo empleado en cada paso del proceso de modelado.

5.2. Nube de puntos sintética

La nube de puntos sintética ha sido generada a partir de elementos representativos del castillo y contiene varios muros y torres (Figura 5.1a). El modelo es relativamente simple en términos de diversidad de categorías; principalmente está formado por los elementos estructurales estudiados. Se ha generado una nube de un millón de puntos para todo el modelo utilizando fragmentos de los datos originales. La segmentación de la nube generó 31 conjuntos de datos diferentes y la parametrización final contiene 20 planos y 11 cilindros (ver Figura 5.1b, Figura 5.1c).

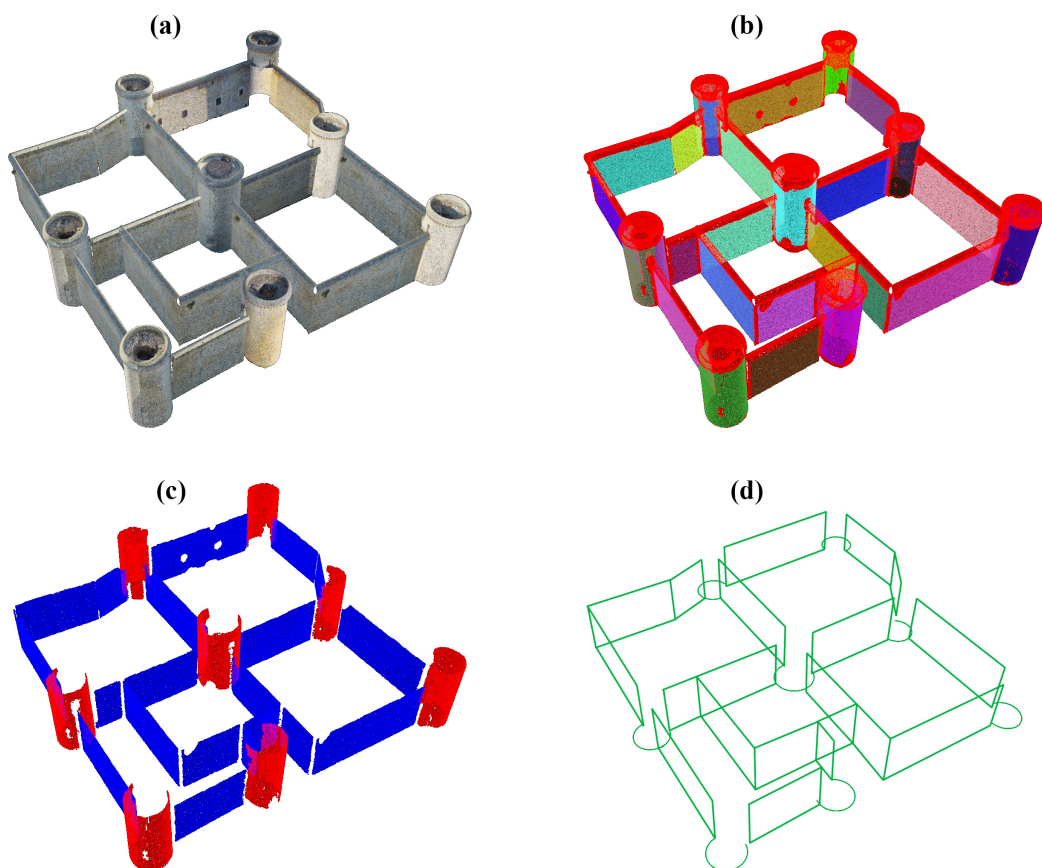


FIGURA 5.1: Nube sintética. a) Preprocesado; b) Segmentado; c) Clasificado; d) Modelo final.

Tal y como se muestra en la Tabla 5.2, la metodología propuesta es verdaderamente exitosa a la hora de detectar los muros y las torres que contiene la nube de puntos y todos los segmentos son reconocidos correctamente en términos de su categoría asignada. En relación a la precisión del modelo, Leronés y otros [17] afirman que una resolución geométrica de 15mm durante la adquisición de la nube de puntos es adecuada para este tipo de aplicaciones y que incluso resoluciones inferiores pueden ser inmanejables en

función del tamaño del modelo y del ordenador donde se realice el tratamiento de la nube capturada.

Teniendo en cuenta estas cifras, se ha asumido que un error durante el modelado del orden de magnitud de unos pocos centímetros puede ser aceptable para un 3D de estas características. Teniendo en cuenta estas consideraciones, se puede afirmar que la precisión geométrica del modelo, representada por la distancia media de los puntos al modelo ajustado (ver Tabla 5.2), es suficientemente preciso ya que los planos cuentan con una distancia media global de 24 ± 1 mm mientras que los cilindros cuentan con una distancia media ligeramente superior de 64 ± 2 mm que puede ser explicada por la mayor complejidad del proceso de parametrización de este tipo de primitivas, pero que siguen estando dentro del rango de distancias propuesto.

Por otra parte, el modelo generado aún cuenta con algunos errores, que puede deberse a limitaciones de la metodología propuesta. En primer lugar, tres torres fueron divididas en dos segmentos (marcadas con * en la Tabla 5.2) y registradas por tanto como dos torres diferentes. En segundo lugar, el algoritmo de intersección no fue capaz de detectar una de las intersecciones entre un plano y uno de los cilindros. Debe destacarse que los algoritmos de que consta la metodología propuesta dependen en parte de la selección manual de parámetros y que, por lo tanto, se ven afectados por el error humano. Sin embargo, estos inconvenientes no afectan al resultado global ya que la etapa de parametrizado es suficientemente robusta como para extraer las principales características de las primitivas detectadas incluso en presencia de información incompleta, tal y como se muestra en la Figura 5.1d.

Segmento	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
Plano %	100	10	99.7	12.6	100	11.7	100	100	100	12.7	10	99.6	46.5	13.3	100	
Cilindro %	14.2	2.7	0.0	100	0.0	100	0.0	2.4	3.0	98.3	23.4	24.3	100	100	13.9	
μ (mm)	44.5	23.2	19.5	63.9	22.5	69.8	23.2	21.9	21.0	67.9	23.4	20.1	56.6	64.7	25.0	
σ (mm)	1.0	0.4	0.7	2.1	0.4	2.6	0.4	0.3	0.3	1.9	0.4	0.6	1.3	2.2	0.6	
Nota													1*			
Segmento	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Plano %	100	100	100	100	30.8	22.3	19.1	71.9	38.3	100	100	100	99.3	99.8	100	12.4
Cilindro %	26.4	35.2	2.4	18.2	100	100	100	100	99.4	11.1	12.1	17.4	22.0	18.0	19.7	100
μ (mm)	19.9	24.4	20.2	22.9	55.3	60.5	62.6	52.4	81.0	22.7	20.9	24.2	31.5	22.0	20.3	70.9
σ (mm)	0.3	0.5	0.2	0.4	1.2	1.8	1.8	0.9	3.2	0.3	0.3	0.5	1.0	0.3	0.3	3.0
Nota					3*	1*	2*	2*	3*							

TABLA 5.2: Nube de puntos sintética. Porcentaje de puntos ajustados, distancia media al modelo y desviación estándar.

Segmento	1	2	3	4	5	6	7
Plano %	81	10	11	83	100	12	59
Cilindro %	0	91	93	25	46	99	36
Categoría	P	C	C	P	P	C	U
μ (mm)	42	72	62	48	28	67	-
σ (mm)	1.6	2.2	2.5	1.6	0.8	2.0	-

TABLA 5.3: Nube de puntos del Castillo de Torrelobatón. Porcentaje de puntos ajustados, distancia media al modelo ajustado y desviación estándar.

5.3. Nube de puntos real

El fragmento de la nube de puntos del Castillo de Torrelobatón cuenta con 177.397 puntos, contiene tres torres y tres muros y ha sido seleccionada para probar el desempeño del algoritmo con condiciones reales de trabajo. Durante la primera fase, el algoritmo de preprocesamiento disminuyó el tamaño de la nube alrededor de un 40 %. El proceso de segmentación fue capaz de extraer siete segmentos diferentes a partir de la nube preprocesada y 36.957 puntos de 101.520 (36.40 %) fueron dejados sin segmentar.

En base a los resultados obtenidos, se ha establecido de forma experimental que el número de puntos ajustados por el modelo debe ser de al menos un 75 % para que un segmento pertenezca a una de las categorías mencionadas. Una vez establecido este requisito, el algoritmo de clasificación fue capaz de clasificar de forma correcta los tres muros de la nube como planos, las tres torres como cilindros y no asignó ninguna categoría a uno de los segmentos al no poder ajustar más de un 60 % de los puntos con ninguno de los dos modelos. En la Tabla 5.3 se recogen los porcentajes de puntos ajustados, además de las distancias medias de éstos y sus respectivas desviaciones estándar a los modelos de ajuste.

Con respecto al algoritmo de parametrización, éste fue capaz de detectar de forma precisa los encuentros entre los muros y las torres. El método de cálculo de interdependencias, las relaciones entre los diferentes elementos detectados y los resultados obtenidos han sido recogidos en la Figura 5.2f.

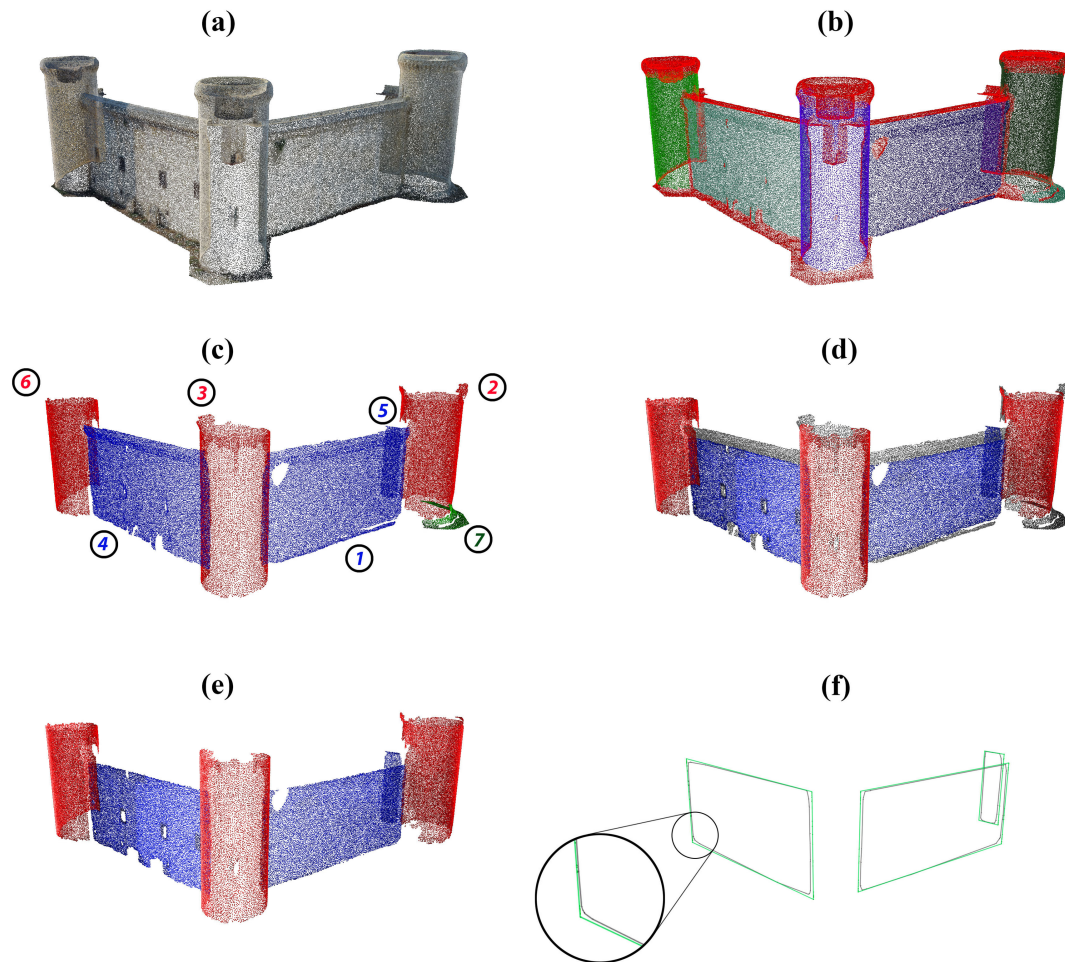


FIGURA 5.2: Nube del castillo. a) Preprocesado; b) Segmentado; c) Clasificado; d) Puntos del modelo y puntos no ajustados en gris; e) Puntos proyectados; f) Parametrización final y bordes mejorados (detalle)

Es posible caracterizar elementos arquitectónicos adicionales de las torres empleando la naturaleza cilíndrica de estos elementos. A partir de la información obtenida durante la parametrización de los cilindros es posible caracterizar las almenas de la torre. El algoritmo propuesto segmenta la nube de puntos del castillo aplicando un filtro que elimina puntos que estén dentro de un rango de valores específicos a lo largo de uno o más ejes de coordenadas. Los puntos que están dentro de un cubo 3D teórico apoyado sobre la parte superior del cilindro son filtrados, siendo el lado del mismo un parámetro relacionado con el valor del radio de la torre. Finalmente, el conjunto de puntos extraído es proyectado sobre un plano que contiene al eje de revolución del cilindro. Los puntos situados en el borde de la proyección se utilizan para definir el perfil del elemento arquitectónico de revolución detectado. Los algoritmos propuestos han sido capaces de extraer correctamente el perfil de las almenas de las torres utilizando los puntos no segmentados tras la

aplicación del algoritmo RGS para optimizar el desempeño computacional. El resultado final ha sido reflejado en la Figura 5.3.

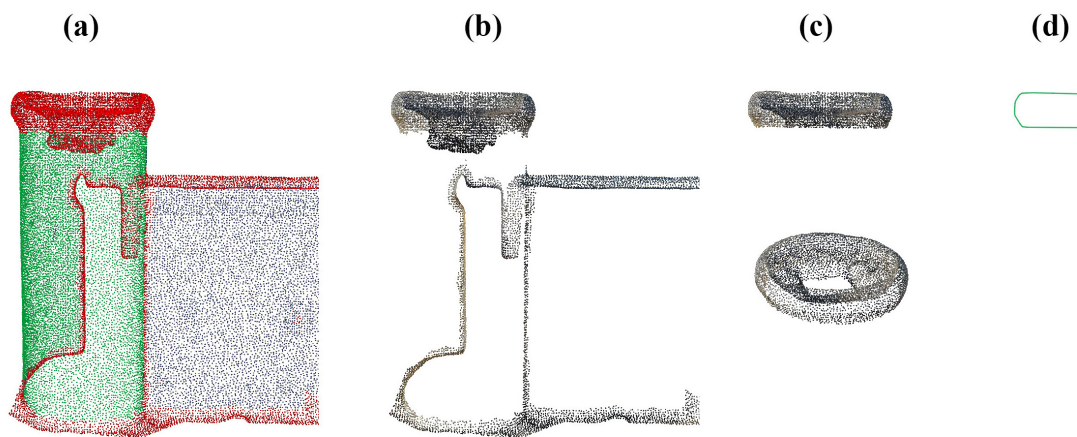


FIGURA 5.3: a) Segmentado; b) No segmentado; c) Proyección de puntos extraídos de revolución; d) Perfil poligonal.

5.4. Modelo BIM generado

Los elementos arquitectónicos parametrizados por la herramienta son automáticamente generados en Revit. En la Figura 5.4a se ha representado la nube de puntos real del Castillo de Torrelobatón. El modelo BIM final generado en Revit una vez finalizado el procesamiento de la nube ha sido representado en la Figura 5.4b. Como se puede apreciar, el modelo sólido recoge los elementos geométricos más representativos presentes en la nube de puntos inicial.

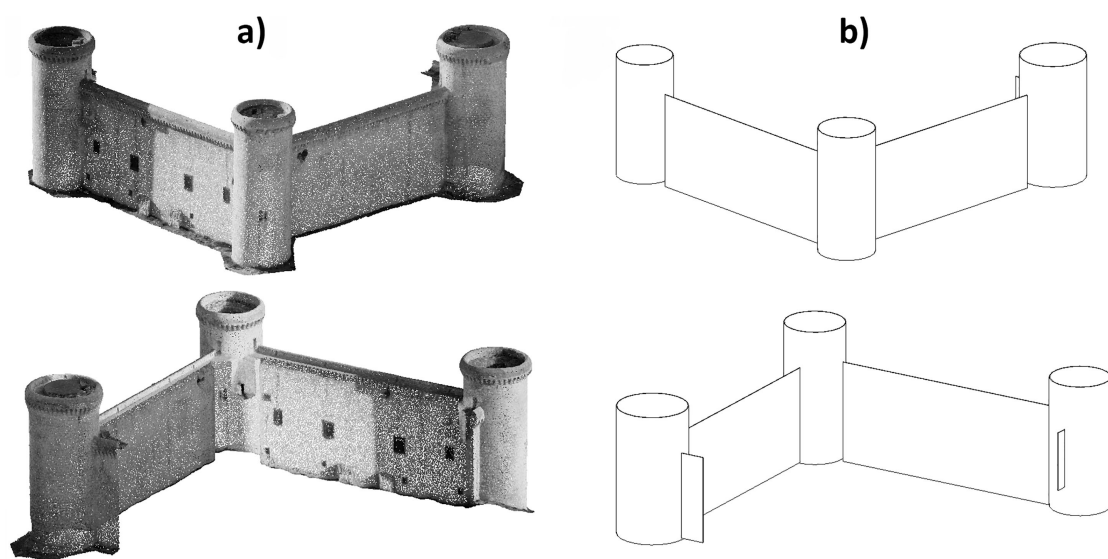


FIGURA 5.4: Modelado BIM en Revit. a) Nube de puntos de partida; b) Modelo final generado automáticamente.

Capítulo 6

Estudio económico

En este capítulo se presenta un estudio teórico del coste aproximado que ha supuesto el desarrollo y realización de este proyecto. Se presentan tanto los costes directos, derivados del salario del personal y de los equipos y material utilizado, como los costes indirectos.

6.1. Costes directos

6.1.1. Coste de personal

La realización del proyecto ha sido llevada a cabo por una única persona encargada tanto del análisis previo como del desarrollo e implementación de la aplicación software. Para calcular el coste asociado al trabajo realizado se ha supuesto que esta persona es un ingeniero que se rige por la tabla salarial del convenio del metal. El salario cobrado por esta persona, según el convenio de 2018, ha sido recogido en la Tabla 6.1:

	División	Categoría	Salario	Plus	Salario	Salario	Hora
Grupo	Funcional	Profesional	Base	Convenio	Total/mes	Annual	Extra
1	Técnicos	Ingenieros	807,86	1.255,42	2.063,28	28.885,92	20,26

TABLA 6.1: Tabla salarial del convenio del metal.

Si se tiene en cuenta la aportación del 23.6 % que realiza la empresa a la Seguridad Social, el coste anual bruto de la mano de obra de un ingeniero se correspondería con el que ha sido reflejado en la Tabla 6.2.

Coste anual	
Salario bruto	28,885.92 €
Seguridad Social	6,817,07 €
Coste total	35,702.99 €

TABLA 6.2: Coste anual para la empresa de un ingeniero.

Teniendo en cuenta que en 2018 hubo 251 días laborales y suponiendo una jornada laboral de 7,5 horas al día, se obtiene que durante esta anualidad un empleado trabajó 1.882 horas. El coste bruto por hora de un ingeniero para la empresa se ha calculado como la división del salario bruto anual entre las horas efectivas trabajadas:

$$\frac{\text{coste}}{\text{hora}} = \frac{35.702,99 \frac{\text{€}}{\text{año}}}{1.882 \frac{\text{horas}}{\text{año}}} = 18,97 \frac{\text{€}}{\text{hora}} \quad (6.1)$$

En la Tabla 6.3 se muestra una estimación de las horas empleadas y su distribución en función de la tarea realizada durante el desarrollo del proyecto:

Estimación de horas empleadas	
	Horas
Análisis	240 horas
	Lectura y análisis de artículos 160 horas
	Análisis de bibliotecas para visión 80 horas
Cursos	240 horas
	Curso de C++ 120 horas
	Curso sobre librería PCL 80 horas
	Curso de Visual Studio 40 horas
Diseño	320 horas
	Diseño de algoritmos 200 horas
	Diseño de estructura de clases 120 horas
Implementación	300 horas
	Implementación de algoritmos 200 horas
	Corrección de errores 100 horas
Documentación	240 horas
	Algoritmos 80 horas
	Redacción de memoria 120 horas
	Redacción de artículo de congreso 40 horas
Total	1340 horas

TABLA 6.3: Distribución temporal de trabajo

El coste del personal ha sido calculado multiplicando el número de horas empleadas por el coste de una hora de trabajo:

$$1.340 \text{ horas} \cdot 18,97 \frac{\text{€}}{\text{hora}} = 25.419,8 \text{ €} \quad (6.2)$$

6.1.2. Coste de amortización de equipos

En la Tabla 6.4 se analizan los costes de la amortización de los equipos de trabajo utilizados. Se ha supuesto que el periodo de amortización del equipo informático, que coincide con su vida útil, es de 6 años.

El coste final por hora de utilización del material se calcula como la división de la amortización anual entre el número de horas de uso de los equipos.

	Total	Anual
Ordenador HP Pavilion	950,00 €	158,33 €
Total material	950,00 €	158,33 €

TABLA 6.4: Coste asociado al material.

$$\frac{\text{coste final}}{\text{hora}} = \frac{158,33 \frac{\text{€}}{\text{año}}}{1882 \frac{\text{horas}}{\text{año}}} \approx 0,08 \frac{\text{€}}{\text{hora}} \quad (6.3)$$

Para calcular el coste asociado a la utilización del equipo se ha supuesto que éste ha sido utilizado durante la realización de todo el proyecto. El coste de amortización del material se calcula según la siguiente expresión:

$$1340 \text{ horas} \cdot 0,08 \frac{\text{€}}{\text{hora}} = 107,20 \text{ €} \quad (6.4)$$

6.1.3. Coste de amortización de software

En la Tabla 6.5 se analizan los costes de la amortización del software utilizado. Se ha supuesto que el periodo de amortización de software es de 4 años.

	Total	Anual
Windows 10 Pro	49,00 €	12,25 €
Revit 2018	3.043,15 €	760,79 €
Total	3.092,15 €	773,04 €

TABLA 6.5: Coste asociado a las aplicaciones software.

Los programas informáticos no incluidos en la tabla anterior son de libre distribución y su coste, por tanto, es nulo.

6.1.4. Coste de material fungible

El coste del material fungible empleado en el proyecto ha sido reflejado en la Tabla 6.6.

	Precio	Unidades	Total
Impresión	3,00 €	4,00 €	12,00 €
Encuadernación	20,00 €	4,00 €	80,00 €
Total			92,00 €

TABLA 6.6: Material fungible

6.1.5. Costes directos totales

Los costes directos totales se obtienen como la suma de los costes de personal, la amortización de equipos y de software y el coste de los consumibles.

$$Coste = 25.419,80 + 107,20 + 773,04 + 92,0 = 26.392,04 \text{ €} \quad (6.5)$$

6.2. Coste indirecto

Los costes indirectos se producen por la actividad requerida para la elaboración del proyecto y han sido recogidos en la Tabla 6.7.

Costes indirectos	
Dirección y servicios administrativos	90,00 €
Consumo de electricidad	120,00 €
Consumo de telefonía	15,00 €
Consumo de desplazamiento	30,00 €
Total gastos indirectos	255,00 €

TABLA 6.7: Costes indirectos

6.3. Coste total

Los costes totales se obtienen tras sumar los gastos directos y los indirectos y han sido recogidos en la Tabla 6.8.

Costes totales	
Costes directos	26.392,04 €
Costes indirectos	255,00 €
Coste total del proyecto	26.617,04 €

TABLA 6.8: Coste total del proyecto

Capítulo 7

Conclusiones y líneas futuras

En este capítulo se exponen los logros técnicos conseguidos durante el desarrollo del proyecto. Además, se presentan las aplicaciones prácticas reales del desarrollo realizado y las conclusiones a las que se ha llegado tras analizar los resultados obtenidos. Finalmente, se exponen las futuras líneas de desarrollo que pueden llevarse a cabo para mejorar el desempeño de los algoritmos desarrollados.

7.1. Conclusiones

A lo largo del presente trabajo fin de máster se han completado con éxito los diferentes objetivos propuestos. Se ha desarrollado correctamente una nueva metodología para generar modelos paramétricos de forma automática de los muros y de las torres de edificaciones históricas. La nueva metodología propuesta es capaz de, utilizando nubes de puntos como material de partida, determinar los elementos constituyentes básicos de la edificación así como estimar sus dimensiones aplicando sucesivos algoritmos en un proceso multietapa.

Adicionalmente, durante el transcurso del proyecto se han completado con éxito los siguientes objetivos:

- Se ha recopilado una extensa biblioteca de artículos sobre la reconstrucción tridimensional de edificaciones empleando técnicas de digitalización 3D. Para ello ha sido necesario estudiar el funcionamiento de los sistemas de gestión documental orientada a artículos académicos como es Mendeley.
- Se ha desarrollado con éxito un algoritmo de clasificación que de forma automática clasifica y etiqueta los elementos arquitectónicos más representativos de nubes de puntos. El algoritmo ha sido optimizado para la detección de muros y torres de castillos.
- Se ha desarrollado con éxito un algoritmo de parametrización que de forma automática determina las dimensiones de los elementos detectados dentro de la nube de puntos. De forma adicional, el algoritmo es capaz de detectar la conectividad local entre los planos y los cilindros analizados.
- Se han implementado y testeado los algoritmos propuestos en lenguaje C++. Para ello ha sido necesario profundizar en el conocimiento del dicho lenguaje de programación. Además, ha sido necesario desarrollar código fuente legible dividido en ficheros y bibliotecas utilizando software de desarrollo profesional (Visual Studio).
- Se ha integrado la biblioteca de visión artificial PCL en el desarrollo realizado para robustecer, simplificar y mejorar la calidad general del código escrito. Para ello ha sido necesario analizar en profundidad las funcionalidades que ofrece esta librería de tratamiento de nubes de puntos.

- Se ha integrado el código desarrollado en una biblioteca dinámica (DLL) para su fácil integración en otros proyectos. Para ello ha sido necesario comprender el funcionamiento de este tipo de archivos así como analizar las ventajas e inconvenientes que supone su utilización.
- Se ha desarrollado un plug-in para la aplicación Revit para que los algoritmos de procesamiento desarrollados puedan ser utilizados directamente desde un entorno profesional BIM mejorando de esta forma su usabilidad.
- Se ha probado el desempeño de los algoritmos propuestos. Para analizar el comportamiento del algoritmo se han utilizado dos nubes de puntos diferentes que contienen elementos arquitectónicos representativos.

Una vez completado el desarrollo de la herramienta se puede afirmar que las aplicaciones prácticas del desarrollo son elevadas, especialmente para la generación automática de modelos básicos de edificaciones. La detección tanto de muros como torres hace de ésta una herramienta única en su género. Sin embargo, la detección de paredes planas puede ser utilizada tanto para edificios históricos, como castillos o palacios, como para edificaciones de nueva construcción ya que presentan una elevada cantidad de elementos constructivos de este tipo.

7.2. Líneas futuras

7.2.1. Ampliación del clasificador

El conjunto básico de primitivas detectadas, compuesto por planos y cilindros, puede ser ampliado agregando otras primitivas geométricas para analizar elementos más avanzados como torres cubiertas o cúpulas utilizando un enfoque similar al propuesto que aumente el conjunto de características arquitectónicas detectadas por la herramienta.

7.2.2. Detección de aperturas

El modelo final generado podría verse enriquecido al implementar un sistema de detección de aperturas. Varios autores han realizado desarrollos capaces de detectar tanto puertas como ventanas en fachadas de bloques de edificios principalmente. Sin embargo,

la detección de este tipo de aberturas en fachadas de edificios históricos es aún un tema a resolver, tanto para fachadas planas como con superficies curvas o irregulares.

Bibliografía

Referencias por orden de aparición en el texto.

- [1] J Adami, F Guerra, and P Vernier. Laser scanner and architectural accuracy test. In *Proceedings of CIPA 2007*, Athens (Greece), October 1–6 2007. ResearchGate. XXI International Symposium.
- [2] S Antonopoulou, D Arch, and P Bryan. *BIM for Heritage. Developing a Historic Building Information Model*. Historic England, Model, Swindon, 2017.
- [3] Y Arayici, J Counsell, L Mahdjoubi, G Nagy, S Hawas, and K Dweidar, editors. *Heritage Building Information Modelling*. Routledge, UK, 1 edition, 2017.
- [4] L Barazzetti, F Banfi, R Brumana, M Previtali, and F Roncoroni. Bim from laser scans not just for buildings: Nurbs-based parametric modeling of a medieval bridge. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume III-5, 2016.
- [5] V Barrile, G Bilotta, D Lamari, and G. M Meduri. Comparison between techniques for generating 3d models of cultural heritage. In *Recent Advances in Mechanics, Mechatronics and Civil, Chemical and Industrial Engineering, Mathematics and Computers in Science and Engineering Series*, volume 49, pages 140–145, 2015.
- [6] S Bruno, M Fino, and F Fatiguso. Historic building information modelling: performance assessment for diagnosis-aided information modelling and management. *Automation in Construction*, 86:256–276, 2018.
- [7] CIPA. Cipavila 2019 27th international symposium. <https://www.cipa2019.org/>, 2019. [Accedido 1-Mayo-2019].

- [8] CloudCompare. 3d point cloud and mesh processing software open source project. <https://www.danielgm.net/cc/>, 2019. [Accedido 18-Mayo-2019].
- [9] Cyclone. Software de procesamiento de nubes de puntos 3d. <https://leica-geosystems.com/es-es/products/laser-scanners/software/leica-cyclone>, 2019. [Accedido 18-Mayo-2019].
- [10] Universidad de Granada. Digitalización 3d. <https://bit.ly/2LSWRj6>, 2019. [Accedido 7-Enero-2019].
- [11] Historic England. Bim for heritage: Developing a historic building information model. <https://bit.ly/2BMLGJV>, 2017. [Accedido 25-Abril-2019].
- [12] D G. Aguilera and J G. Lahoz. Laser scanning or image-based modeling? a comparative through the modelization of san nicolas church. *International Archives of Photogrammetry and Remote Sensing*, 36(B5), 2012.
- [13] Inception. Inception eu-project. <https://www.inception-project.eu/en>, 2019. [Accedido 18-Enero-2019].
- [14] Y Jaidar et al. Digitalización tridimensional para la documentación, análisis y conservación de bienes culturales: los relieves decorativos en piedra de la zona arqueológica de tula, hidalgo, méxico. *Photogrammetry and Remote Sensing*, 8(16): 43–56, 2017.
- [15] D Laefer. New city-scale damage prediction methods during urban tunnelling. In *Structural Analysis of Historical Constructions*, pages 15–22. CRC Press/Balkema, 2016.
- [16] Leica. Leica cloudworks para revit. <https://bit.ly/2LSLJHB>, 2019. [Accedido 7-Enero-2019].
- [17] Pedro M. Leronés, José Llamas, Álvaro Gil, Jaime Gómez García Bermejo, and Eduardo Zalama Casanova. A practical approach to making accurate 3d layouts of interestin cultural heritage sites through digital models. *Elsevier, Journal os Cultural Heritage*, 11:1–9, 2010.
- [18] P.M Leronés et al. *Heritage BIM models from 3D pointclouds*, chapter 3.3, pages 1–73. INCEPTION consortium, 2018.

- [19] Z Li, L Zhang, P Mathiopoulos, F Liu, S Li, and H Liu. A hierarchical methodology for urban façade parsing from tfs poin clouds. *Photogrammetry and Remote Sensing*, 123:75–93, 2017.
- [20] F López, P Lerones, J Llamas, J García-Bermejo, and E Zalama. A framework for using point cloud data of heritage buildings towards geometry modeling in a bim context: a case study on santa maria la real de mave church. *International Journal of Architectural Heritage*, 11(7):965–986, 2017.
- [21] F López, P Lerones, J Llamas, J García-Bermejo, and E Zalama. Semi-automatic generation of bim models for cultural heritage. *International Journal of Heritage Architecture*, 2(2):293–302, 2018.
- [22] F.J. López, P Martín Lerones, J Llamas, J Gómez-García-Bermejo, and E Zalama. Linking hbim graphical and semantic information through the getty aat: Practical application to the castle of torrelatón. *Proceedings of the Heri-Tech International Conference*, May 2018a. Florence (Italy).
- [23] F.J López, P Martín Lerones, J Llamas, J Gómez-García-Bermejo, and E Zalama. A review of heritage building information modeling (h-bim). *Multimodal Technologies Interaction*, 2(2):21, 2018b. [<http://www.mdpi.com/2414-4088/2/2/21>].
- [24] F Maietti, R Giulio, M Balzani, E Piaia, M Medici, and F Ferrari. 3d data acquisition and modelling of complex heritage buildings. *Digital cultural heritage*, 02.1, May 23–25 2017. Final conference of the Marie Sklodowska-Curie initial training network, Olimje (Slovenia).
- [25] P Martín Lerones, J Llamas Fernández, A Melero Gil, J Gómez García-Bermejo, and E Zalama Casanova. A practical approach to making accurate 3d layouts of interesting cultural heritage sites through digital models. *Journal of Cultural Heritage*, 11:1–9, 2010.
- [26] P Martín Lerones, D Olmedo Vélez, F Gayubo Rojo, J Gómez-García-Bermejo, and E Zalama. Moisture detection in heritage buildings by 3d laser scanning. *Studies in conservation*, 61:46–54, April 2016. doi: 10.1179/2047058415Y.0000000017.
- [27] MeshLab. Meshlab.net. <http://www.meshlab.net>, 2019. [Accedido 18-Mayo-2019].

- [28] S Ochmann, R Vock, R Wessel, and R Klein. Automatic reconstruction of parametric building models from indoor point clouds. *Computers and Graphics*, 54: 94 – 103, 2016. doi: <https://doi.org/10.1016/j.cag.2015.07.008>. Special Issue on CAD/Graphics 2015.
- [29] PCL. Point cloud library. <http://pointclouds.org/>, 2019. [Accedido 18-Mayo-2019].
- [30] M Previtali and F Banfi. Towards the definition of work flows for automation in hbim generation. In *Digital Heritage. Progress in Cultural Heritage*, volume 1, pages 52–63. Springer International Publishing, 2018.
- [31] Recap. Create accurate 3d models with reality capture. <https://www.autodesk.com/products/recap/overview>, 2019. [Accedido 18-Mayo-2019].
- [32] J Román Cembranos, P Martín Leronés, et al. Supporting the automatic extraction of hbim elements from point clouds. In *Digital Heritage. Progress in Cultural Heritage*, volume 2, pages 3–10. Springer International Publishing, 2018. doi: 10.1007/978-3-030-01765-1_1.
- [33] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.
- [34] J San José Alonso, J Martínez-Rubio, J Fernandez, and J García Fernandez. Comparing time-of and phase-shift the survey of the royal pantheon in the basilica of san isidoro (león). *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVIII-5/W16:377–385, March 2–4 2012. doi: 10.5194/isprsarchives-XXXVIII-5-W16-377-2011. Trento (Italy).
- [35] A Stambler and D Huber. Building modeling through enclosure reasoning. In *International Conference on 3D Vision Workshops*, pages 118–125. IEEE, 2015. doi: 10.1109/3DV.2014.65.
- [36] C Tommasi, C Achille, and F Fassi. From point cloud to bim: A modelling challenge in the cultural heritage field. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XLI-B5, 2016.

-
- [37] A Walford. A new way to 3d scan (eos systems). <http://www.ee.co.za/wp-content/uploads/legacy/PositionIT%202009/A%20new%20way%20to%203D%20scan.pdf>. [Accedido 20-Enero-2019].
- [38] C Wang, Y Cho, and C Kim. Automatic bim component extraction from point clouds of existing buildings for sustainability applications. *Automation in Construction*, 56: 1–13, 2015.
- [39] X Yang, M Koehl, and P Grussenmeyer. Mes-to-bim: From segmented mesh elements to bim model with limited parameters. *ISPRS - The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 1213–1218, 2018. doi: 10.5194/isprs-archives-XLII-2-1213-2018.
- [40] E Özdemir and F Remondino. Segmentation of 3d photogrammetric point cloud for 3d building modeling. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4/W10:135–142, 09 2018. doi: 10.5194/isprs-archives-XLII-4-W10-135-2018.
- [41] S Zolanvari and D Laefer. Slicing method for curved façade and window extraction from point clouds. *Photogrammetry and Remote Sensing*, 119:334–346, 2016.