



POLITÉCNICA

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES
UNIVERSIDAD POLITÉCNICA DE MADRID

José Gutiérrez Abascal, 2. 28006 Madrid
Tel.: 91 336 3060
info.industriales@upm.es

www.industriales.upm.es

Diego Munoz De Escalona

05 TRABAJO FIN DE MASTER

INDUSTRIALES

TRABAJO FIN DE MASTER

VALIDACIÓN GEOMÉTRICA EN IMPRESIÓN 3D MEDIANTE VISIÓN RGB-D

Febrero 2026

Diego Munoz De Escalona

DIRECTORES DEL TRABAJO FIN DE MASTER:

Daniel Galán Vicente

Javier Laserna Moratalla



POLITÉCNICA



UNIVERSIDAD
POLITÉCNICA
DE MADRID



Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros Industriales
Máster Universitario en Automática y Robótica



Trabajo Fin de Máster
Validación Geométrica en Impresión 3D mediante Visión
RGB-D

Autor: Diego Munoz De Escalona

Director 1: Daniel Galán Vicente
Profesor Permanente Laboral

Director 2: Javier Laserna Moratalla
Director Externo

enero, 2026

*A mi familia y amigos, por su apoyo constante.
Gracias por estar siempre.*

Agradecimientos

En primer lugar, quiero agradecer a mis padres, Antonio Muñoz De Escalona y Vincenza Tripi, por el apoyo incondicional que me han brindado y por hacer posible que pudiera cursar esta maestría. Gracias por la confianza, el esfuerzo y el acompañamiento constante que han sostenido cada etapa de este camino.

Agradezco también a mi familia, por estar siempre presente y por animarme en los momentos difíciles. Su cercanía y sus palabras de aliento han sido un impulso fundamental para seguir adelante.

De manera especial, deseo expresar mi agradecimiento a mis tutores y profesores, Daniel Galán y Javier Laserna, por su guía durante la investigación, por compartir su conocimiento y por facilitar los materiales y recursos necesarios para llevar el proyecto a término. Su orientación y apoyo han sido determinantes para el desarrollo de este trabajo.

Asimismo, agradezco a mis amigos y compañeros de máster, Sergio, Fiorella y David, por la ayuda durante todo el curso, por las horas de estudio compartidas, por los trabajos realizados en equipo y por los buenos momentos vividos en la universidad.

Por último, agradezco a mis amigos Sergio, Daniel, Amanda, Azocar, Sonia y Alejandra, por el apoyo constante a lo largo de este proceso y por los ánimos que me han dado para continuar y alcanzar mis objetivos.

Resumen

La fabricación aditiva por deposición de filamento (Modelado por Deposición Fundida (FDM)) se ha consolidado como una alternativa accesible para la producción de prototipos y piezas funcionales. Sin embargo, la calidad final puede verse afectada por defectos que aparecen durante la impresión y que no siempre se corrigen ajustando parámetros del proceso. En este contexto, esta investigación propone un sistema de validación geométrica capaz de estimar el estado de avance de una impresión 3D comparando la geometría observada durante la fabricación con un modelo teórico de referencia.

El enfoque se basa en la adquisición de información tridimensional mediante un sensor Cámara de Profundidad (RGB-D), a partir del cual se capturan pares sincronizados de color y profundidad para reconstruir nubes de puntos de la escena. Con el fin de garantizar repetibilidad entre capturas, la nube observada se expresa en un marco de referencia estable asociado a la celda de impresión, permitiendo comparar sesiones aun cuando la cámara cambie de posición u orientación. Posteriormente, se aplica una etapa de pre-procesamiento orientada a reducir ruido, eliminar puntos fuera de rango y delimitar la región de interés, de modo que la información relevante de la pieza quede preservada y sea comparable con el modelo.

Sobre estas nubes, se implementa un flujo de registro 3D para alinear la nube adquirida con la nube teórica, priorizando robustez frente a solapamientos parciales, oclusiones y correspondencias espurias, condiciones habituales en escenarios capa a capa. Finalmente, se definen métricas cuantitativas de progreso e integridad geométrica que permiten interpretar el estado de la impresión desde perspectivas complementarias, proporcionando un criterio reproducible para evaluar la evolución del proceso.

Los resultados muestran que la disponibilidad de geometría condiciona la estabilidad del registro y la fiabilidad de las métricas. Con pocas capas impresas, geometría parcial, el sistema es más sensible al ruido y a fallos de alineamiento, mientras que en niveles de

impresión intermedios y altos la convergencia del registro se estabiliza y la estimación del progreso se vuelve consistente. En conjunto, el sistema desarrollado constituye una herramienta modular y extensible para el control de calidad geométrico en impresión 3D basada en nubes de puntos, sentando una base para futuras mejoras orientadas a seguimiento temporal, fusión multivista y criterios automáticos de aceptación/rechazo.

Palabras clave: impresión 3D, fabricación aditiva, RGB-D, nube de puntos, registro 3D, control de calidad geométrico, estimación de progreso, RealSense, ArUco.

Códigos UNESCO: 1203.05 (Sistemas automatizados de producción), 1203.06 (Sistemas automatizados de control de calidad), 1203.25 (Diseño de sistemas sensores), 2209.94 (Visión Artificial, Análisis de Imágenes y Vídeo),

Abstract

Additive manufacturing by fused deposition modeling (FDM) has become an accessible alternative for producing prototypes and functional parts. However, final quality may be affected by defects that arise during printing and are not always corrected by tuning process parameters. In this context, this research proposes a geometric validation system capable of estimating the progress state of a 3D print by comparing the geometry observed during fabrication with a theoretical reference model.

The approach relies on acquiring three-dimensional information using an RGB-D sensor, from which synchronized color and depth pairs are captured to reconstruct point clouds of the scene. To ensure repeatability across acquisitions, the observed cloud is expressed in a stable reference frame associated with the printing cell, enabling comparisons across sessions even when the camera changes position or orientation. A preprocessing stage is then applied to reduce noise, remove out-of-range points, and delimit the region of interest, so that the relevant part geometry is preserved and remains comparable to the model.

On these point clouds, a 3D registration workflow is implemented to align the acquired cloud with the theoretical one, prioritizing robustness under partial overlaps, occlusions, and spurious correspondences—conditions commonly encountered in layer-by-layer scenarios. Finally, quantitative metrics of progress and geometric integrity are defined to interpret the printing state from complementary perspectives, providing a reproducible criterion to assess process evolution.

The results show that the amount of available geometry strongly influences registration stability and metric reliability. With only a few layers printed and therefore partial geometry, the system is more sensitive to noise and alignment failures, whereas at intermediate and advanced printing stages the registration convergence stabilizes and progress estimation becomes consistent. Overall, the developed system constitutes a modular and

extensible tool for point-cloud-based geometric quality control in 3D printing, laying the groundwork for future improvements oriented toward temporal tracking, multi-view fusion, and automatic accept/reject criteria.

Keywords: 3D printing, additive manufacturing, RGB-D, point cloud, 3D registration, geometric quality control, progress estimation, RealSense, ArUco.

UNESCO codes: 1203.05 (Automated production systems), 1203.06 (Automated quality control systems), 1203.25 (Sensor system design), 2209.94 (Computer Vision, Image and Video Analysis).

Índice

Índice de Figuras	xx
Índice de Tablas	xxi
Lista de Símbolos	xxi
Lista de Acrónimos	xxiv
1. Introducción	1
1.1. Contexto general	1
1.1.1. Tendencia en la fabricación aditiva	2
1.1.2. Proceso de impresión 3D	5
1.2. Planteamiento del problema	6
1.3. Motivación	7
1.4. Objetivos	8
1.4.1. Objetivo general	8
1.4.2. Objetivos específicos	8
1.5. Contribución del trabajo	9
1.6. Estructura del documento	9
2. Estado del arte	11
2.1. Técnicas de adquisición de información 3D del mundo real	11

2.2.	Sensores RGB-D para reconstrucción cercana	13
2.3.	Calibración intrínseca, extrínseca y color profundidad	16
2.3.1.	Componentes de la calibración en sistemas RGB-D	16
2.3.2.	Calibración intrínseca del canal RGB e IR	16
2.3.3.	Calibración extrínseca y alineación color profundidad	16
2.3.4.	Fundamentos teóricos del modelo de cámara	18
2.3.5.	Reconstrucción 3D	18
2.4.	Preprocesamiento de nubes de puntos	19
2.5.	Registro de nubes de puntos y alineación con el modelo teórico	20
2.6.	Monitoreo y control superficial en fabricación aditiva a partir de datos 3D .	21
2.7.	Síntesis del estado del arte	22
3.	Metodología	25
3.1.	Objetivo y enfoque general	25
3.2.	Entorno hardware y configuración experimental	26
3.3.	Entorno software y herramientas de desarrollo	27
3.3.1.	Sistema operativo y configuración base	27
3.3.2.	Librerías del software	28
3.4.	Estructura modular del proyecto	28
3.5.	Estrategia de desarrollo y validación	29
3.5.1.	Desarrollo incremental por módulos	29
3.5.2.	Validación por etapas	30
3.5.3.	Relación con el diseño experimental	30
4.	Arquitectura del sistema de adquisición y análisis de nubes de puntos	33
4.1.	Adquisición de datos	34
4.1.1.	Normalización geométrica de la escena	34
4.1.2.	Delimitación de la región de interés	37
4.1.3.	Arquitectura software del módulo	37
4.1.4.	Fusión de nubes de puntos	43

4.2.	Representación geométrica	44
4.2.1.	Fundamentos teóricos	44
4.2.2.	Arquitectura software del módulo	46
4.3.	Preprocesamiento de nubes	50
4.3.1.	Fundamentos teóricos	50
4.3.2.	Arquitectura software del módulo	53
4.4.	Registro de nubes de puntos	57
4.4.1.	Fundamentos teóricos del registro	58
4.4.2.	Arquitectura software del módulo	60
4.5.	Evaluación y comparación	66
4.5.1.	Fundamentos teóricos	66
4.5.2.	Arquitectura software del módulo	69
4.6.	Visualización	73
4.6.1.	Fundamentos teóricos	73
4.6.2.	Arquitectura software del módulo	74
4.7.	Interfaz gráfica de usuario	77
4.7.1.	Flujos de uso principales	77
4.7.2.	Detalles de implementación	78
5.	Resultados y discusión	81
5.1.	Diseño de los experimentos y conjunto de datos	81
5.1.1.	Escenarios de progresión y poses de cámara	83
5.2.	Desempeño del algoritmo en el registro de nubes de puntos	84
5.2.1.	Visión global del registro para una pose	84
5.2.2.	Comparación por nivel de progresión	86
5.2.3.	Visión global del registro para dos poses fusionadas	88
5.3.	Métricas de progreso de impresión	91
5.3.1.	Progreso en altura: <code>zmax_percent</code>	91
5.3.2.	Cobertura superficial: <code>progress_surface</code>	96
5.3.3.	Coincidencia volumétrica: <code>perc_b_in_a</code>	100

5.3.4. Coherencia entre métricas y lectura conjunta	105
6. Conclusión y Trabajo Futuro	107
6.1. Síntesis del trabajo realizado	107
6.2. Cumplimiento de los objetivos planteados	108
6.3. Limitaciones del sistema	109
6.4. Líneas de trabajo futuras	110
6.5. Conclusión final	111
7. Bibliografía	113
A. Diagrama UML de los módulos del programa	119
B. Análisis de aplicación e impacto	127
B.1. Aplicaciones	127
B.2. Evaluación de impactos	128
B.2.1. Impacto social	128
B.2.2. Impacto económico	129
B.2.3. Impacto medioambiental	129
B.2.4. Otros impactos relevantes	130
B.3. Análisis de aspectos éticos	130
B.3.1. Privacidad y protección de datos	130
B.3.2. Fiabilidad, riesgo de error y responsabilidad	130
B.3.3. Transparencia, interpretabilidad y trazabilidad	131
B.3.4. Propiedad intelectual y licencias	131
B.4. Contribución a los Objetivos de Desarrollo Sostenible	131
B.5. Marco normativo de aplicación	132
B.6. Estándares aplicables y contribuciones	133
B.7. Síntesis	134
C. Planificación temporal y presupuesto	135
C.1. Estructura de Paquetes de Trabajo	135

C.2. Planificación	137
C.3. Presupuesto	139
C.3.1. Costo de los materiales	139
C.3.2. Amortización de los equipos	139
C.3.3. Costo del personal	139
C.3.4. Costo total del proyecto	140

Índice de Figuras

1.1. Evolución del porcentaje de publicaciones en robótica relacionadas con impresión 3D en Web of Science (2013–2023).	3
1.2. Evolución del porcentaje de publicaciones en robótica relacionadas con impresión 3D en SCOPUS (2013–2023).	4
1.3. Evolución del porcentaje de publicaciones en robótica relacionadas con impresión 3D en IEEE Xplore (2013–2023).	4
1.4. Diagrama de bloques de impresión 3D por FDM.	5
2.1. Representación de una nube de puntos	13
3.1. Montaje de los componentes	26
3.2. Pieza impresa apilable	27
4.1. Arquitectura general del sistema de adquisición y análisis de nubes de puntos	34
4.2. Ejemplo de capturas del sistema durante la adquisición.	39
4.3. Nube de puntos generada de las capturas 4.2a y 4.2b	40
4.4. Resultado final de la adquisición	42
4.5. Fusión de nubes de puntos a partir de dos poses distintas de la cámara. . .	43
4.6. Nube de puntos generada a partir de un modelo <code>.stl</code>	48
4.7. Nube de puntos adquirida experimentalmente	48
4.8. Proyecciones ortogonales de la nube tras la normalización geométrica. . . .	49
4.9. Flujo de preprocesamiento para nube de puntos generada experimentalmente	55

4.10. Flujo de preprocesamiento para nube de puntos generada de un archivo .stl	57
4.11. Registro de las nubes de puntos con el algoritmo CliReg	63
4.12. Registro de las nubes de puntos con el algoritmo RANSAC	64
4.13. Registro de las nubes de puntos con el algoritmo Registro Global Rápido (FGR)	64
4.14. Visualización de proyecciones	76
4.15. Interfaz gráfica del sistema.	78
4.16. Flujo de trabajo para la interfaz de la Figura 4.15b	79
4.17. Flujo de trabajo para la interfaz de la Figura 4.15c	79
5.1. Nivel 1: Sección baja de la pieza impresa	82
5.2. Nivel 2: Sección baja + media de la pieza impresa	82
5.3. Nivel 3: Sección completa de la pieza impresa	82
5.4. Adquisición en la pose 1 de la cámara.	83
5.5. Adquisición en la pose 2 de la cámara.	83
5.6. Adquisición fusionada.	84
5.7. Solapamiento para el registro del nivel bajo de la pieza.	84
5.8. Solapamiento para el registro del nivel bajo y medio de la pieza.	85
5.9. Solapamiento para el registro de la pieza completa.	85
5.10. Solapamiento para el registro del nivel bajo con captura de dos poses diferentes.	88
5.11. Solapamiento para el registro del nivel bajo+medio con captura de dos poses diferentes.	89
5.12. Solapamiento para el registro de la pieza completa con captura de dos poses diferentes.	89
5.13. Distribución de <code>zmax_percent</code> para el registro del nivel bajo de la pieza con una única pose de cámara.	92
5.14. Distribución de <code>zmax_percent</code> para el registro del nivel bajo+medio de la pieza con una única pose de cámara.	92
5.15. Distribución de <code>zmax_percent</code> para el registro de la pieza completa con una única pose de cámara.	93

5.16. Distribución de <code>zmax_percent</code> para el registro del nivel bajo a partir de la fusión de dos poses de cámara.	93
5.17. Distribución de <code>zmax_percent</code> para el registro del nivel bajo+medio a partir de la fusión de dos poses de cámara.	94
5.18. Distribución de <code>zmax_percent</code> para el registro de la pieza completa a partir de la fusión de dos poses de cámara.	94
5.19. Distribución de <code>progress_surface</code> para el nivel bajo de la pieza con una única pose de cámara.	96
5.20. Distribución de <code>progress_surface</code> para el nivel bajo+medio de la pieza con una única pose de cámara.	97
5.21. Distribución de <code>progress_surface</code> para la pieza completa con una única pose de cámara.	97
5.22. Distribución de <code>progress_surface</code> para el nivel bajo a partir de la fusión de dos poses de cámara.	98
5.23. Distribución de <code>progress_surface</code> para el nivel bajo+medio a partir de la fusión de dos poses de cámara.	98
5.24. Distribución de <code>progress_surface</code> para la pieza completa a partir de la fusión de dos poses de cámara.	99
5.25. Distribución de <code>perc_b_in_a</code> para el registro del nivel bajo de la pieza con una única pose de cámara.	101
5.26. Distribución de <code>perc_b_in_a</code> para el registro del nivel bajo+medio de la pieza con una única pose de cámara.	101
5.27. Distribución de <code>perc_b_in_a</code> para el registro de la pieza completa con una única pose de cámara.	102
5.28. Distribución de <code>perc_b_in_a</code> para el registro del nivel bajo a partir de la fusión de dos poses de cámara.	102
5.29. Distribución de <code>perc_b_in_a</code> para el registro del nivel bajo+medio a partir de la fusión de dos poses de cámara.	103
5.30. Distribución de <code>perc_b_in_a</code> para el registro de la pieza completa a partir de la fusión de dos poses de cámara.	103
A.1. Módulo de adquisición de datos	120
A.2. Módulo de representación geométrica	121
A.3. Módulo de preprocesamiento de nubes	122

A.4. Modulo del Pipeline 123

A.5. Módulo de registro de nubes de puntos 124

A.6. Módulo de evaluación y comparación 125

A.7. Módulo de visualización 126

C.1. Paquetes de trabajo del proyecto 136

C.2. Diagrama de Gantt del proyecto 138

Índice de Tablas

4.1. Clases del módulo de adquisición: etapa, responsabilidades y ecuaciones. . .	42
4.2. Clases del módulo de representación geométrica: etapa, responsabilidades y ecuaciones.	49
4.3. Resumen de operaciones del preprocesamiento: etapas, responsabilidades y criterios aplicados.	56
4.4. Resumen de operaciones del módulo de registro: etapas, responsabilidades y criterios geométricos.	65
4.5. Resumen de operaciones del módulo de evaluación y comparación: etapas, responsabilidades y criterios aplicados.	72
4.6. Resumen de operaciones del módulo de visualización: etapas, responsabilidades y criterios aplicados.	76
C.1. Costo de los materiales del proyecto	139
C.2. Amortización de los equipos	139
C.3. Costo del personal del proyecto	140
C.4. Costo total del proyecto	140

Lista de Acrónimos

- ABS** Acrilonitrilo Butadieno Estireno. 1
- CAD** Diseño Asistido por Computadora. 1, 5–7
- FDM** Modelado por Deposición Fundida. vii, xvii, 1, 3, 5–7, 86, 127, 133
- FGR** Registro Global Rápido. xviii, 57, 59–61, 63–65
- FOV** Campo de Visión. 14
- GUI** Interfaz Gráfica de Usuario. 77
- IR** Infrarrojo. 12, 16
- JSON** Notación de Objetos JavaScript. 37, 41, 53
- KDTree** Árbol k-dimensional. 75
- NN** Vecino Más Cercano. 66, 67, 70
- PETG** Tereftalato de Polietileno Glicol. 1
- PLA** Ácido Poliláctico. 1
- PLY** Formato de Archivo Polygon. 41, 77
- PnP** Perspectiva- n -Puntos. 17
- RGB-D** Cámara de Profundidad. vii, ix, 8, 11, 13–17, 20
- RMSE** Raíz del Error Cuadrático Medio. 67, 70, 72

SDK Kit de Desarrollo de Software. 38

SfM Estructura a partir del Movimiento. 11, 12, 22

SLA Estereolitografía. 2, 6

SLS Sintetizado Selectivo por Láser. 2

STL Estereolitografía. 5

ToF Tiempo de Vuelo. 12, 15, 22

TPU Poliuretano Termoplástico. 1

UML Lenguaje Unificado de Modelado. 41, 46, 48, 53, 60, 69, 74, 75

Introducción

Este capítulo ofrece una visión de la impresión 3D como fabricación aditiva. En conjunto, este marco sitúa la impresión 3D no solo como prototipado, sino como un conjunto de procesos con impacto real en producción.

1.1. Contexto general

La impresión 3D, también denominada fabricación aditiva, es un proceso de manufactura en el que una pieza se construye de forma incremental mediante la adición controlada de material a partir de un modelo digital. El sistema se encarga de generar la geometría final siguiendo trayectorias y secciones definidas por el modelo del Diseño Asistido por Computadora (CAD). La diferencia entre tecnologías se explica principalmente por el estado del material de entrada, filamento sólido, resina líquida o polvo, y por el mecanismo físico de solidificación, extrusión y enfriamiento o fusión por láser, lo que condiciona la resolución alcanzable, el acabado superficial y las propiedades mecánicas. [1], [2]

Actualmente, existen tres tecnologías representativas:

- FDM: utiliza como material de aporte un filamento termoplástico que puede ser Ácido Poliláctico (PLA), Acrilonitrilo Butadieno Estireno (ABS), Tereftalato de Polietileno Glicol (PETG) o Poliuretano Termoplástico (TPU), incluyendo variantes compuestas con cargas o fibras. El filamento se funde en el extrusor y se deposita a través de una boquilla en cordones sucesivos que se adhieren al material previamente depositado, formando cada capa por trayectorias. Su principal ventaja técnica es la accesibilidad y versatilidad, debido al bajo coste relativo del equipo, la disponibilidad de materiales y la rapidez para iterar prototipos y piezas funcionales. Su calidad final

depende de parámetros acoplados como temperatura de boquilla y cama, altura de capa, velocidad de impresión, caudal, orientación de las trayectorias y estrategia de relleno, que afectan la adhesión entre capas, la anisotropía mecánica, la rugosidad superficial y la aparición de deformaciones térmicas. [3]

- Estereolitografía (SLA): emplea una resina fotopolimérica líquida como material de entrada y solidifica selectivamente cada capa mediante una fuente de luz, típicamente ultravioleta, que activa la reacción de curado. Su ventaja técnica es la alta resolución y el buen acabado superficial, lo que permite reproducir detalles finos y geometrías pequeñas con mayor fidelidad que otros procesos de polímeros. Como contrapartida, suele requerir operaciones de posprocesado como lavado para eliminar resina no curada, poscurado para estabilizar propiedades y retirada de soportes, que pueden influir en marcas superficiales y tolerancias dimensionales. [4]
- Sintetizado Selectivo por Láser (SLS): utiliza polvo como material de aporte y un láser sinteriza selectivamente el material para consolidar cada capa. Una ventaja técnica clave es que el propio polvo no consolidado actúa como soporte durante la fabricación, reduciendo la necesidad de soportes específicos y habilitando geometrías complejas. Además, permite fabricar piezas funcionales con buena resistencia y repetibilidad, a costa de un equipo más complejo y de requisitos operativos asociados al manejo del polvo, la limpieza de la pieza y un acabado superficial que puede requerir postprocesado según la aplicación. [5], [6]

1.1.1. Tendencia en la fabricación aditiva

La investigación en fabricación aditiva ha experimentado un crecimiento durante la última década. Este incremento no solo responde a un aumento global del número de publicaciones, sino también a una mayor integración de estas tecnologías dentro de áreas consolidadas como la robótica. En este contexto, se ha realizado un análisis bibliométrico entre 2013 y 2023 centrado en el porcentaje de publicaciones del ámbito de la robótica que abordan la impresión 3D, lo que permite evaluar su relevancia dentro del campo.

En la base de datos Web of Science se observa una tendencia ascendente en la proporción de trabajos de robótica relacionados con impresión 3D, tanto en artículos de revista como en contribuciones a congresos. A lo largo del periodo analizado, el porcentaje de artículos aumenta de forma progresiva hasta situarse en torno al 3,3 % en 2023, mientras que las publicaciones en congresos alcanzan valores cercanos al 3,0 %. Esta evolución, representada en la Figura 1.1 indica una consolidación de la fabricación aditiva como línea de investigación relevante dentro de la robótica.

Un comportamiento similar se observa en SCOPUS, donde la proporción de publica-

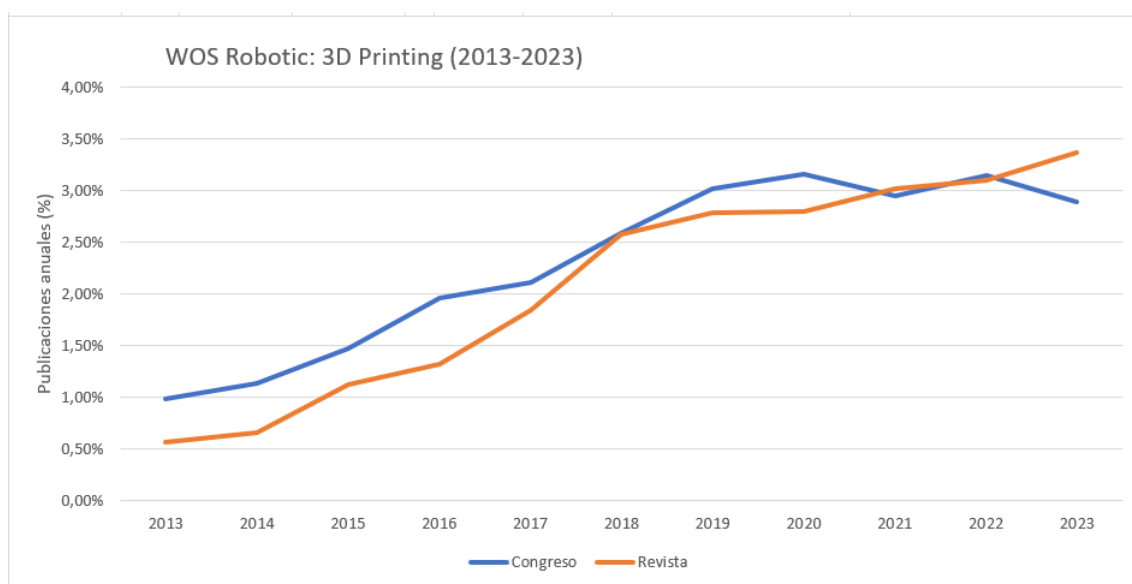


Figura 1.1 Evolución del porcentaje de publicaciones en robótica relacionadas con impresión 3D en Web of Science (2013–2023).

ciones en robótica que tratan sobre impresión 3D presenta un crecimiento durante todo el periodo. En este caso, el aumento resulta especialmente marcado en los últimos años, alcanzando en 2023 valores próximos al 2,7 % en revistas y al 2,9 % en congresos. Esta evolución presentada en la Figura 1.2 sugiere una integración progresiva de la fabricación aditiva en investigaciones de robótica.

En el caso de IEEE Xplore la tendencia resulta similar. El porcentaje de artículos de robótica centrados en impresión 3D muestra un crecimiento continuo, alcanzando valores cercanos al 3,7 % en revistas y al 3,0 % en congresos en 2023. La Figura 1.3 refleja cómo la fabricación aditiva ha ido ganando peso dentro de la robótica.

En conjunto, los tres análisis evidencian que la impresión 3D ha pasado a constituir una línea de investigación con una presencia estable y creciente. Este aumento relativo refleja su transición hacia aplicaciones más maduras, donde la fabricación aditiva interviene directamente en el diseño, fabricación y validación de componentes robóticos funcionales. Factores como el mayor control geométrico, la mejora en la repetibilidad del proceso y la aparición de materiales específicos han ampliado las aplicaciones. [2]

En cuanto a los polímeros, se observa una tendencia hacia compuestos funcionales impresos por FDM y otras familias. Investigaciones recientes describen cómo la optimización de hardware, parámetros y rutas de deposición se combina con nuevos filamentos, tintas y resinas para dirigir la orientación de refuerzos y controlar propiedades térmicas, eléctricas y mecánicas. Este movimiento va de la mano con aplicaciones que integran estructura y función, lo que anticipa líneas futuras con microestructuras más complejas y mejores

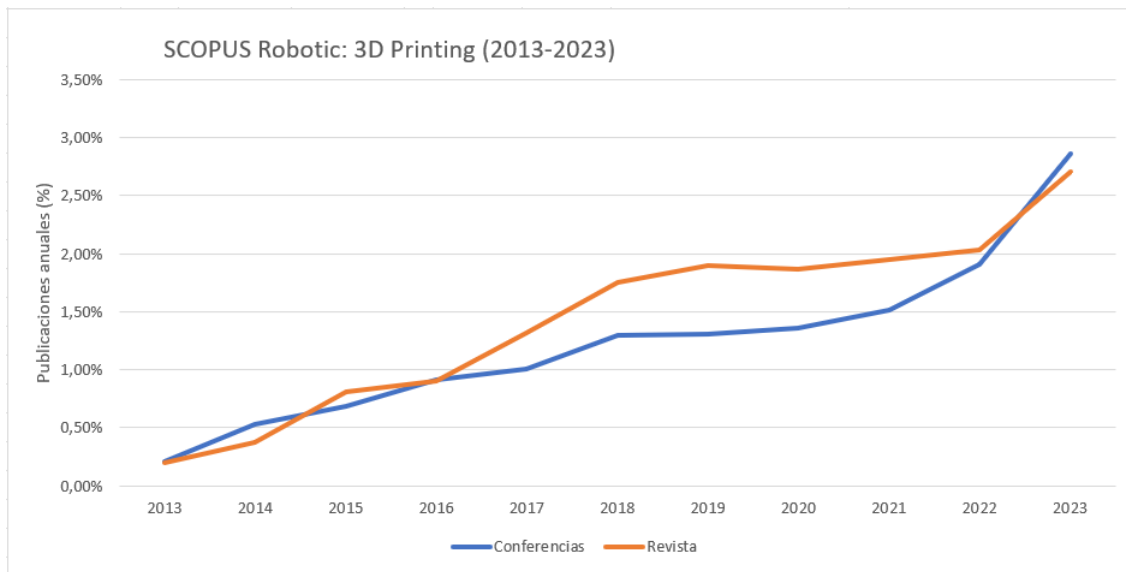


Figura 1.2 Evolución del porcentaje de publicaciones en robótica relacionadas con impresión 3D en SCOPUS (2013–2023).

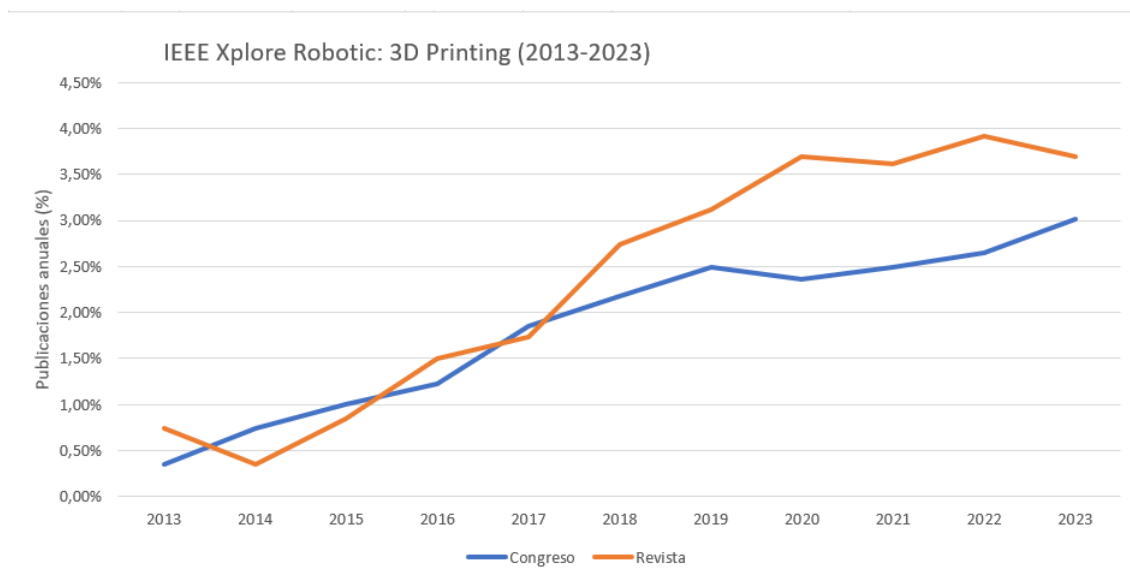


Figura 1.3 Evolución del porcentaje de publicaciones en robótica relacionadas con impresión 3D en IEEE Xplore (2013–2023).

propiedades macroscópicas. [2]

La adopción por sectores sanitarios es un índice de madurez y confianza. En odontología, por ejemplo, las revisiones reportan un crecimiento del uso de impresión 3D en guías quirúrgicas, modelos de planificación y dispositivos personalizados. Esta adopción se apoya en la combinación de escaneo clínico, modelado digital y tecnologías de impresión que reducen tiempos y ajustan el tratamiento al paciente. [4]

También crece la integración de inteligencia artificial en el ciclo aditivo. Las revisiones recientes subrayan el uso de aprendizaje automático para predecir comportamiento de proceso, ajustar parámetros y habilitar monitorización en tiempo real con visión y sensores. El objetivo es cerrar el bucle de control y reducir variabilidad. [7]

Finalmente, varias síntesis de la literatura apuntan a una tendencia a la hibridación con procesos sustractivos y a la formalización de modelos de coste y planificación. Este movimiento reconoce límites actuales en precisión o acabado y propone celdas híbridas donde lo aditivo construye la forma y lo sustractivo depura tolerancias y superficies. En conjunto, estas prácticas están acelerando la transición desde prototipos a producción con criterios de productividad, coste y calidad más claros. [1]

1.1.2. Proceso de impresión 3D

En la Figura 1.4 se presenta un flujo completo de impresión 3D orientado a FDM. Lo primero que se debe de realizar es el modelado de la pieza en CAD un entorno de diseño asistido por computadora. En esta etapa se define la forma exacta de la pieza, materiales de referencia y, si hace falta, restricciones. El resultado es un modelo paramétrico que se exporta a formatos como Estereolitografía (STL) u OBJ para realizar la impresión.

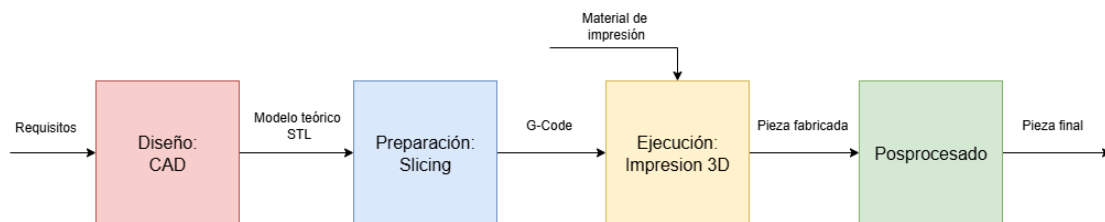


Figura 1.4 Diagrama de bloques de impresión 3D por FDM.

Posteriormente se realiza el slicing, discretizando el volumen en capas. Además, se fijan parámetros que mandan sobre la calidad final como altura de capa, temperaturas, velocidades, relleno y soportes. Para FDM la temperatura del extrusor y de la cama,

sumado a la altura de capa y al ancho de línea, condicionan la adherencia y la estabilidad dimensional. En resina como SLA el slicer define exposición por capa, soportes y deja preparado el poscurado. El objetivo es equilibrar el tiempo de fabricación y la precisión sin comprometer la integridad de la pieza.

Ya con el CAD discretizado se planifican las trayectorias capa a capa para realizar la impresión. Antes de iniciar es conveniente alinear, calibrar y verificar la cama de impresión, comprobar la boquilla del extrusor y asegurar que el filamento esté en buenas condiciones.

Cuando finaliza el proceso de impresión se ejecuta el posprocesado. En caso de una manufactura FDM se retiran soportes y se puede lijar la pieza para reducir el relieve visible en la superficie que dejan las capas. En SLA se lava la pieza en alcohol isopropílico para disolver y arrastrar la resina no polimerizada, dejando la superficie limpia y homogénea.

1.2. Planteamiento del problema

Aunque las impresiones con FDM se ha popularizado por su costo accesible y la facilidad para fabricar piezas complejas, la literatura sigue reportando problemas en la calidad del resultado. Estos problemas se originan en defectos que el propio proceso introduce capa a capa y que no siempre se corrigen al ajustar parámetros de impresión. En consecuencia, aún con perfiles de proceso bien escogidos, la pieza puede desviarse de lo previsto a nivel dimensional y superficial tal y como se define en [3].

Uno de los primeros problemas que se menciona proviene de las tensiones residuales. Al extruir plástico caliente sobre material frío, se producen gradientes de temperatura no uniformes. Ese ciclo rápido de calentar y enfriar genera contracciones que se acumulan como tensiones internas y terminan manifestándose en deformaciones, encorvamiento de esquinas y delaminación entre capas. Esto quiere decir que la pieza intenta despegarse de la base o curvarse en bordes, alterando la geometría planeada. La bibliografía describe estas distorsiones y menciona estrategias típicas para evitar estas problemáticas, como controlar temperatura de boquilla, la cama y reducir la velocidad del extrusor.

Un segundo conjunto de defectos está relacionado con la generación de vacíos. Durante la extrusión del filamento se forman microporos y huecos entre capas, ya sea por encapsulamiento de aire, distribución desigual del material o variación en el diámetro del filamento. Estos vacíos no solo reducen la resistencia, también introducen incertidumbre geométrica porque el borde real de la superficie no coincide exactamente con la envolvente que traza el slicer.

Asociado a lo anterior surge el problema de la adhesión. En plásticos sin refuerzo, la unión entre capas baja cuando el filamento nuevo no se llega a fundir bien con el anterior.

A simple vista se ven líneas de separación entre trazos y, en los ensayos, las fibras pueden desprenderse al romper la pieza. En lo mecánico, la carga no se transfiere bien entre capas.

Por último, en superficies curvas la discretización y textura se perciben a simple vista con crestas y valles. Este acabado limita usos donde el contacto o la estética sean críticos. Además introduce pequeñas discrepancias entre la superficie medida y el modelo CAD que afectan a cualquier estimación geométrica.

En conjunto, la revisión concluye que estos defectos no se eliminan del todo solo con ajustes de parámetros y constituyen una fuente real de variabilidad en piezas FDM, tanto en polímeros como en compuestos.

1.3. Motivación

La impresión 3D se ha consolidado como una tecnología transversal en sectores como salud, automoción, aeroespacial y educación, debido a su capacidad para fabricar geometrías complejas, reducir plazos de prototipado y habilitar estrategias de producción flexible.

Dentro de este panorama, el proceso FDM destaca por su versatilidad, accesibilidad y amplia implementación. Su uso extendido se explica por un equilibrio entre coste, facilidad de operación y capacidad de adaptación a diferentes escenarios. Esta relevancia también se refleja en el interés sostenido por optimizar parámetros de impresión y desarrollar nuevas formulaciones de filamento orientadas a mejorar propiedades mecánicas.

Sin embargo, el crecimiento de FDM viene acompañado de problemáticas bien conocidas. Durante la fabricación pueden aparecer defectos como vacíos, falta de adhesión entre capas, rugosidad superficial y desviaciones dimensionales, que comprometen la calidad final de la pieza. A esto se le suman estrategias de posprocesado que incrementan el tiempo total y no siempre atacan el origen de las desviaciones geométricas. En términos de control de calidad, muchas prácticas siguen siendo reactivas, donde la inspección se realiza al final del proceso, cuando la corrección implica un coste elevado o, directamente, no es viable. Esto motiva el desarrollo de soluciones de monitorización que permitan describir qué se está construyendo realmente mientras la impresión está en curso.

En impresión 3D, el progreso suele calcularse a partir de variables indirectas como el tiempo transcurrido, el número de capas planificadas en el G-code o las estimaciones del slicer. Aunque estas medidas son útiles como referencia, no garantizan que la geometría prevista se haya materializado correctamente en cada instante, ya que no incorporan información sobre desviaciones reales, pérdidas de material u otros fenómenos del proceso. Por ello, se justifica un enfoque geométrico que cuantifique el avance comparando la escena

observada con el modelo teórico de la pieza.

Para sustentar esta medición, se propone un sistema basado en una cámara RGB-D, que proporciona simultáneamente color y profundidad y permite reconstruir nubes de puntos de la escena. En aplicaciones de corto alcance, la selección y configuración del sensor es relevante para limitar errores sistemáticos y reducir la incertidumbre en la reconstrucción. No obstante, disponer de datos 3D no resuelve por sí mismo el problema central: alinear de manera fiable lo observado con lo esperado, especialmente en condiciones frecuentes de impresión por capas como solapamiento parcial, oclusiones y presencia de outliers.

En este contexto, el progreso de impresión suele inferirse mediante variables indirectas como el tiempo transcurrido, el número de capas planificadas en el G-code o las estimaciones del slicer. Aunque son útiles como referencia, estos indicadores no garantizan que la geometría prevista se haya materializado correctamente en cada instante. Por ello, se justifica un enfoque geométrico que cuantifique el avance comparando la escena observada con el modelo teórico de la pieza, proporcionando una medida objetiva, reproducible e interpretable del estado de la impresión.

1.4. Objetivos

1.4.1. Objetivo general

Diseñar, implementar y evaluar un sistema de adquisición y análisis de nubes de puntos basado en un sensor RGB-D que permita comparar la geometría observada durante la impresión 3D con el modelo teórico, proporcionando métricas cuantitativas para validar el progreso de la impresión por capas y la integridad geométrica de la pieza.

1.4.2. Objetivos específicos

1. Definir una arquitectura modular para el sistema de adquisición y análisis de nubes de puntos.
2. Implementar el módulo de adquisición de datos utilizando un sensor RGB-D y normalizar todas las capturas en un marco de referencia común.
3. Desarrollar una etapa de preprocesamiento de nubes de puntos que limpie y estructure los datos obteniendo nubes adecuadas para el registro con el modelo teórico.
4. Integrar y adaptar métodos de registro de nubes de puntos para alinear la nube generada con el modelo, y comparar su comportamiento en distintos niveles de

progresión de la pieza y configuraciones de captura.

5. Definir e implementar métricas específicas de validación geométrica y progreso de impresión que permitan cuantificar el grado de coincidencia entre la impresión real y la geometría teórica.
6. Diseñar y ejecutar una validación experimental del proceso con diferentes niveles de avance de la pieza y estrategias de adquisición para analizar la coherencia de las métricas propuestas como indicadores del estado de la impresión.

1.5. Contribución del trabajo

Este trabajo contribuye con una propuesta para abordar el problema de cuantificar el grado de progresión de una impresión 3D y validar la integridad geométrica de la pieza a medida que se va fabricando. Frente a enfoques basados en indicadores indirectos, tiempo, capas planificadas o estimaciones del slicer, se plantea un criterio de validación basado en la geometría observada, capaz de reflejar de forma objetiva qué parte de la pieza prevista se ha materializado realmente y con qué nivel de coherencia respecto al modelo teórico.

La contribución principal consiste en definir un enfoque de monitorización a partir de información tridimensional. Con esto se busca estimar el avance de la fabricación y detectar desviaciones geométricas mediante métricas interpretables. De este modo, el sistema propuesto proporciona una lectura cuantitativa y reproducible del estado de la impresión, orientada a complementar la inspección tradicional con una evaluación basada en la forma real construida en cada instante.

Adicionalmente, el trabajo aporta un sistema modular del proceso, de manera que la adquisición de datos, la alineación con el modelo de referencia y el cálculo de métricas se integran en un flujo coherente. Esta organización facilita la adaptación del sistema a distintos escenarios de captura y a diferentes geometrías, manteniendo como objetivo central la evaluación del progreso y la integridad geométrica de la impresión durante su ejecución.

1.6. Estructura del documento

A continuación, se describe de forma resumida la organización del documento, indicando el contenido principal de cada capítulo.

1. **Capítulo 1. Introducción.** Presenta el contexto de la impresión 3D y la necesidad de validar la geometría durante el proceso de fabricación. Define el problema, la

motivación, los objetivos y la contribución del trabajo, además de guiar la lectura del resto del documento.

2. **Capítulo 2. Estado del arte.** Revisa técnicas de adquisición 3D y el uso de sensores RGB-D, incluyendo aspectos de calibración, modelo de cámara y reconstrucción. Sintetiza enfoques de preprocesamiento, registro y monitoreo geométrico en fabricación aditiva que sustentan las decisiones del sistema propuesto.
3. **Capítulo 3. Metodología.** Describe el enfoque general de diseño y validación del sistema, así como el procedimiento experimental seguido. Detalla el entorno de trabajo, hardware y software, y los criterios utilizados para estructurar el pipeline y definir el esquema de evaluación.
4. **Capítulo 4. Arquitectura del sistema de adquisición y análisis de nubes de puntos.** Expone la arquitectura modular del sistema: adquisición, representación geométrica, preprocesamiento, registro y evaluación. Presenta el flujo de datos entre módulos e incluye los componentes de visualización e integración del sistema completo.
5. **Capítulo 5. Resultados y discusión.** Define los experimentos realizados y analiza el desempeño del registro bajo distintos escenarios y estrategias de captura. Evalúa métricas de progreso y discute su utilidad para estimar el estado de la impresión.
6. **Capítulo 6. Conclusión y trabajo futuro.** Resume los principales aportes y el grado de cumplimiento de los objetivos planteados a partir de los resultados obtenidos. Identifica limitaciones del sistema y propone líneas de mejora y trabajo futuro para robustecer el enfoque presentado.

Estado del arte

En este capítulo se revisan los trabajos previos que sustentan el diseño e implementación de un sistema de adquisición y análisis de nubes de puntos para la validación geométrica en impresión 3D por capas. En particular, se resume el estado del arte en sensores RGB-D para reconstrucción cercana, los procedimientos de calibración necesarios para obtener medidas métricas consistentes y las etapas típicas de preprocesamiento de nubes de puntos. Finalmente, se analizan enfoques de registro 3D y estrategias de monitoreo orientadas a estimar progreso y detectar desviaciones geométricas, delimitando así las decisiones metodológicas adoptadas en este trabajo.

2.1. Técnicas de adquisición de información 3D del mundo real

La adquisición de información 3D a partir de escenas reales se apoya en un conjunto de tecnologías ópticas y de visión por computador que estiman distancia. Estos enfoques se clasifican principalmente en métodos pasivos, que infieren profundidad a partir de imágenes sin iluminar activamente la escena, y métodos activos, donde el sistema proyecta o emite luz para facilitar la estimación métrica. Esta clasificación es común en revisiones de sensores 3D y resulta útil para anticipar resolución, rango, robustez y coste computacional. [8]

En los métodos pasivos destacan la estereovisión y la fotogrametría multivista. En ambos casos la geometría se recupera triangulando correspondencias entre imágenes, ya sea entre dos cámaras estéreo o entre múltiples vistas, Estructura a partir del Movimiento (SfM) . Su principal ventaja es que pueden alcanzar alta resolución espacial y buena

cobertura angular. Sin embargo, su desempeño se degrada en superficies con poca textura, repetitivas o con brillos, y suelen requerir múltiples vistas para lograr reconstrucciones estables. En aplicaciones de metrología ligera y levantamiento 3D, SfM se ha consolidado como una alternativa de bajo coste para reconstruir topografía y superficies complejas a distintas escalas. [9]

Los métodos activos se apoyan en principios de medición más directos. En triangulación activa el sistema proyecta un patrón y estima profundidad por el desplazamiento observado en la cámara. Típicamente ofrece buena precisión a corta distancia, pero es sensible a iluminación ambiente, oclusiones y materiales reflectivos. El análisis de sensores basados en triangulación reporta que el error crece con la distancia y que la reflectividad y la saturación en Infrarrojo (IR) pueden inducir huecos y outliers, lo que hace necesaria una ventana de trabajo bien definida y una calibración cuidadosa. [10]

Otra familia activa es el Tiempo de Vuelo (ToF), donde la distancia se estima a partir del retardo entre la señal emitida y la recibida. Las cámaras ToF permiten rangos mayores y capturas rápidas, pero están sujetas a sesgos en presencia de superficies complejas. Una revisión discute estas ventajas y limitaciones, así como procedimientos de calibración y fusión con otros sensores. [11] En comparativas experimentales entre tecnologías, se observa que luz estructurada y ToF presentan perfiles de error diferentes. Mientras la primera tiende a fallar en brillos y sombras del proyector, ToF es especialmente sensible a superficies reflectivas, por lo que la elección depende del escenario y del nivel de detalle requerido. [12]

Además del principio físico empleado, un requisito crítico en la adquisición es la calibración del sistema. La geometría reconstruida sólo es interpretable en términos métricos si se conocen los parámetros de cámara y distorsión, llamados parámetros intrínsecos, y si la relación espacial entre sensores o vistas está correctamente modelada con los extrínsecos. En sensores que generan mapas de profundidad es frecuente la presencia de sesgos sistemáticos dependientes de la distancia, del material o del modo de captura, lo que hace necesario caracterizar y corregir dichos efectos para evitar que el error se propague a la nube de puntos y al registro 3D. [10], [11], [13], [14]

En escenarios multivista o de múltiples sensores, la calibración es aún más crítica porque la reconstrucción fusiona observaciones desde diferentes poses o dispositivos. La literatura muestra que pequeñas inconsistencias en los intrínsecos y extrínsecos degradan la coherencia del modelo y se manifiestan como una desviación en los puntos. Por ello, se han propuesto procedimientos de calibración para sistemas RGB-D orientados a reconstrucción completa de escenas, donde se asegura una consistencia entre capturas. [14], [15]

En el contexto de monitoreo geométrico en impresión 3D por capas, estas diferen-

cias tecnológicas y la calibración son determinantes. La escena suele observarse a corta distancia y se busca detectar desviaciones pequeñas capa a capa. Esto conlleva a que se interesa priorizar alta densidad local y repetibilidad temporal, manteniendo el mínimo error posible ante iluminación, reflectividad del material y oclusiones. Este razonamiento motiva el uso de sensores RGB-D de corto alcance y en estrategias de adquisición y preprocesamiento que estabilicen la medida antes del registro 3D con el modelo teórico. [8], [11], [12]

En el monitoreo geométrico de impresión 3D por capas, la elección de la tecnología de medida y la calibración condicionan la comparabilidad entre capturas. Al operar a corta distancia y buscar desviaciones pequeñas capa a capa, se requiere alta densidad de puntos, estabilidad temporal y control del error ante variaciones de iluminación, reflectividad y oclusiones. Bajo estas exigencias, es necesario usar sensores RGB-D de corto alcance y en estrategias de adquisición y preprocesamiento que reduzcan ruido. Sobre esto, la siguiente sección discute sensores RGB-D representativos para reconstrucción cercana, y el apartado posterior desarrolla los fundamentos de calibración intrínseca, extrínseca y color profundidad necesarios para asegurar consistencia métrica durante toda la adquisición. [8], [11], [12]

2.2. Sensores RGB-D para reconstrucción cercana

Un sensor RGB-D es un dispositivo muy utilizado en robótica, que captura de forma sincronizada una imagen a color y un mapa de profundidad para cada píxel, tal que cada uno de los puntos de la escena tiene registrado un color y distancia en el mismo instante. Al fusionar ambos canales, color y profundidad, se genera una nube de puntos; un conjunto discreto de muestras con coordenadas en (x, y, z) que sirve como una representación geométrica de una escena. En la Figura 4.3 se puede observar una nube de puntos, [16], [17]



Figura 2.1 Representación de una nube de puntos

La profundidad del píxel se obtiene según el principio físico de cada dispositivo. En cámaras de luz estructurada como la Intel RealSense SR300 la distancia se calcula a partir de un patrón infrarrojo que se proyecta sobre la escena, lo que permite buen detalle a corta distancia, aunque con sensibilidad al movimiento y materiales reflectivos. Su rango óptimo está limitado a distancias cortas, alrededor de 1,5 metros, y su estabilidad mecánica y óptica permite mantener la calibración frente a variaciones térmicas y vibraciones. [18]

En la familia Intel RealSense D400 se emplea la visión estéreo con iluminación activa, compuesto por dos sensores sincronizados y un proyector infrarrojo que texturiza la escena. Esto favorece la resolución local en rangos cercanos, pero depende de la textura y de la geometría. Dentro de esta familia se encuentran los modelos D415, D435i y D455, donde el D415 destaca en precisión a corta distancia gracias a su menor campo de visión y mayor densidad de píxeles, lo que lo hace especialmente adecuado para reconstrucciones cercanas. El D435i y el D455 ofrecen mayores campos de visión y rangos de captura extendidos hasta 6 metros, pero con un incremento en el error sistemático a distancias cortas. [19]

El modelo Microsoft Azure Kinect DK utiliza el tiempo de vuelo continuo para estimar la fase entre la señal emitida y la recibida, lo que permite derivar la distancia. Este dispositivo es usado para reconstrucciones de larga distancia y en escenas donde pueda variar la iluminación. Por otro lado, esta tecnología genera un suavizado en la imagen por lo que se pierde información de detalles finos y es sensible a superficies reflectivas. [19], [20]

En la impresión 3D por capas la elección del sensor es crucial debido a que la mayoría de las veces se trabaja a corta distancia y cualquier deriva temporal o error acumulado puede deformar cambios reales en la geometría. Un campo de visión pequeño como el del D415 concentra más píxeles en la zona útil y mejora la resolución. Además, dentro de la serie D400 el D415 rinde mejor en reconstrucciones de corto alcance, comparados con los modelos D435i y D455 que ofrecen un Campo de Visión (FOV) más amplios y rangos máximos mayores con menos densidad local. Finalmente, para aplicaciones de control de progreso de impresión mediante registro de nubes de puntos, los sensores basados en luz estructurada o código de luz ofrecen gran detalle y precisión en cortas distancias, siendo adecuados para capturas en cada capa de impresión. Sin embargo, es fundamental considerar las condiciones de iluminación, la reflectividad del material y el rango de trabajo para mitigar errores y mantener la coherencia geométrica durante toda la impresión. [18], [19], [20]

Un punto de referencia es el benchmark SUN RGB-D, que consolidó la evaluación en 3D con 10 335 imágenes RGB-D y anotaciones densas tanto en 2D como en 3D, definiendo tareas y métricas para segmentación, detección y comprensión de escenas completas. El dataset fue capturado con diferentes sensores, entre ellos RealSense, Xtion, Kinect v1

y Kinect v2, lo que permitió estudiar la generalización entre dispositivos y empujar la evaluación hacia espacio 3D en lugar de solo imagen. Para este trabajo aporta dos ideas útiles. Primero, que la medición del desempeño en 3D es más informativa cuando el objetivo es comparar geometría. Segundo, que sensores ligeros y con buen canal de color como RealSense facilitan integraciones en entornos reales sin penalizar la anotación o la evaluación. [21]

Además de las diferencias tecnológicas, un aspecto determinante en reconstrucción cercana es la estructura del error del mapa de profundidad. En sensores de luz estructurada y código de luz, la calidad de la medida depende de la visibilidad del patrón proyectado y de la respuesta óptica del material, por lo que suelen aparecer huecos en bordes, zonas en sombra del proyector y superficies brillantes, así como errores locales en presencia de movimiento. En sensores de visión estéreo activa su incertidumbre crece con la distancia y se degrada en regiones con baja textura. Estas limitaciones justifican restringir la ventana de trabajo y diseñar un preprocesamiento orientado a estabilizar la medida antes del registro. [18], [19]

En robótica, el uso de sensores RGB-D se consolidó porque permite combinar geometría y apariencia para tareas de mapeo y percepción. En particular, en sistemas de mapeo 3D con cámara RGB-D se enfatiza que la reconstrucción depende de la calidad intrínseca de la profundidad y la estabilidad temporal de la adquisición, ya que pequeñas fluctuaciones cuadro a cuadro se traducen en desviaciones geométricas. Esta idea es relevante en impresión por capas, ya que aunque la escena sea estática, el crecimiento progresivo de la pieza implica que el sistema de medida debe ser repetible entre capturas sucesivas para que las diferencias observadas se interpreten como cambios reales y no como ruido del sensor. [16]

Una forma práctica de amortiguar estas fluctuaciones es incorporar filtrado espacial y temporal sobre el canal de profundidad, así como estrategias de fusión que reduzcan el granulado sin borrar discontinuidades geométricas. La literatura en reconstrucción densa en tiempo real muestra que integrar múltiples observaciones produce superficies más estables y mejora la coherencia cuando el sensor introduce ruido y mediciones faltantes. Aunque el objetivo aquí no es reconstruir una malla completa, combinar capturas cercanas y filtrar de manera controlada puede mejorar la consistencia de la nube de puntos previa al registro con el modelo teórico. [20]

Finalmente, desde una perspectiva de diseño del sistema, la elección del sensor RGB-D debe alinearse con las restricciones de la celda de impresión. Distancia corta, posible presencia de vibraciones, variaciones térmicas, oclusiones parciales por la geometría en crecimiento y materiales con reflectividad variable. En este contexto, los sensores de la familia RealSense orientados a corto alcance destacan por su densidad local de puntos y su integración sencilla, mientras que sensores ToF ofrecen mayor rango pero pueden suavizar detalles finos que son críticos cuando se pretende detectar desviaciones pequeñas capa

a capa. En conjunto, estos criterios conectan con el uso de bibliotecas de procesamiento 3D ampliamente adoptadas y con la necesidad de evaluar el desempeño en términos 3D cuando el objetivo final es comparar superficies y dimensiones. [17], [19], [21]

2.3. Calibración intrínseca, extrínseca y color profundidad

Como se ha discutido en la sección anterior, la adquisición 3D sólo puede utilizarse con fines métricos y comparativos si el sistema está correctamente calibrado. En cámaras RGB-D, la calibración permite que el mapa de profundidad y la imagen de color compartan una interpretación geométrica consistente, de modo que cada píxel se traduzca en un punto 3D con coordenadas fiables dentro de un mismo sistema de referencia. [14]

2.3.1. Componentes de la calibración en sistemas RGB-D

En la práctica, la calibración de un sistema RGB-D abarca tres componentes complementarios. Primero, la calibración intrínseca de cada canal, RGB y IR, con la que se estiman la matriz de parámetros internos y la distorsión óptica de cada cámara. Segundo, la calibración extrínseca entre color y profundidad, que determina la transformación rígida que relaciona ambos marcos de referencia. Y tercero, la corrección de sesgos sistemáticos del mapa de profundidad, asociados al principio de medida, a la distancia de trabajo y a las propiedades reflectivas del material. [14]

2.3.2. Calibración intrínseca del canal RGB e IR

Para estimar los intrínsecos del canal RGB, el método flexible de Zhang [13] con patrón planar es una referencia consolidada. Observando un tablero en diferentes orientaciones, el procedimiento permite recuperar la matriz de calibración y los coeficientes de distorsión con buena repetibilidad, incluso con lentes moderadamente distorsionadas. En sistemas RGB-D, es habitual aplicar un procedimiento análogo sobre el canal infrarrojo asociado a la profundidad, de manera que ambos canales queden calibrados bajo un marco geométrico equivalente y comparable.

2.3.3. Calibración extrínseca y alineación color profundidad

La calibración extrínseca entre color y profundidad determina la transformación rígida, rotación y traslación, que permite expresar puntos 3D obtenidos desde el sensor de

profundidad en el sistema de coordenadas de la cámara RGB. Esto es necesario porque, aunque el dispositivo se denomine RGB-D, en realidad integra dos subsistemas ópticos con centros de proyección y geometrías propias. En términos prácticos, esta relación se estima haciendo observable un mismo patrón en ambos canales y calculando la pose relativa mediante técnicas geométricas, comúnmente formuladas como un problema Perspectiva- n -Puntos (PnP). [18], [19]

Además, la consistencia de la extrínseca depende de la sincronización temporal entre la captura de color y la de profundidad. Si ambas no se adquieren simultáneamente, pequeñas vibraciones o movimientos pueden introducir discrepancias que se interpretan como errores geométricos. En sensores como Intel RealSense SR300 y D415, la sincronización por hardware se mantiene en el orden de los milisegundos, lo que favorece estimaciones reproducibles de la relación color profundidad durante sesiones prolongadas de adquisición. [18], [19]

En entornos de producción, algunos sensores incorporan mecanismos para mantener la calibración frente a variaciones térmicas y deriva mecánica. En particular, se reportan esquemas de compensación térmica y procedimientos de calibración en fábrica orientados a estabilizar el acoplamiento entre proyector, canal IR y cámara RGB. Para aplicaciones de corto alcance, estas características resultan relevantes porque ayudan a explicar por qué ciertos dispositivos conservan consistencia métrica durante adquisiciones largas o en entornos con pequeñas perturbaciones. [18], [19]

Para fijar un marco de referencia estable en una celda de trabajo o recuperar la pose del sensor cuando se desplaza, los marcadores fiduciales como ArUco y AprilTag son una solución ampliamente usada. Estos permiten estimar poses de seis grados de libertad con tolerancia a oclusiones parciales y cambios de iluminación, lo que facilita anclar el sistema de coordenadas a la estructura de la impresora y reducir la variabilidad entre sesiones. En la práctica, esto disminuye la varianza de la estimación y contribuye a que las diferencias observadas entre nubes correspondan a cambios reales en la geometría. [22]

En resumen, un flujo típico de calibración en investigación comienza con la calibración intrínseca del canal RGB, continúa con la calibración del canal asociado a la profundidad y la estimación de la transformación rígida entre ambos canales, y finalmente incorpora un mecanismo de referencia en la celda para asegurar repetibilidad entre capturas. Este encadenado proporciona una base métrica consistente para construir nubes de puntos comparables y aplicar etapas posteriores de registro y evaluación geométrica.

2.3.4. Fundamentos teóricos del modelo de cámara

El proceso de calibración se apoya en el modelo pinhole de la cámara. Este explica la relación geométrica entre los puntos ubicados en un espacio de tres dimensiones y sus proyecciones en un plano. La cámara es considerada como un orificio diminuto que proyecta los rayos de luz sobre un plano, generando una relación entre coordenadas tridimensionales (X, Y, Z) y coordenadas bidimensionales (u, v) . [13]. Esta relación se linealiza y se expresa a través de la matriz de calibración intrínseca \mathbf{K} y permite la conversión de las coordenadas del espacio de cámara a coordenadas de píxel con la siguiente relación:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}, \quad \text{donde} \quad \mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.1)$$

En esta matriz, f_x y f_y son las distancias focales en píxeles sobre los ejes horizontal y vertical; mientras que c_x y c_y indican las coordenadas del centro óptico proyectado en el plano de la imagen. La escala s relaciona las coordenadas del plano de imagen con las coordenadas reales del punto tridimensional. [13]

Los parámetros contenidos en \mathbf{K} son propios de cada cámara y se determinan mediante un proceso de calibración interna. En el caso de la cámara Intel RealSense estos valores pueden obtenerse directamente a través de la librería RealSense SDK 2.0.

2.3.5. Reconstrucción 3D

Una vez conocidos los intrínsecos, la reconstrucción 3D se realiza aplicando la retro-proyección a cada píxel (u, v) con profundidad Z :

$$X = \frac{(u - c_x)}{f_x} Z, \quad Y = \frac{(v - c_y)}{f_y} Z, \quad (2.2)$$

donde (X, Y, Z) son las coordenadas del punto en el sistema de la cámara. Este cálculo aplicado sobre todos los píxeles válidos genera una nube de puntos que describe la superficie observada. Sin embargo, su precisión depende de la estabilidad del sensor y de la calidad de la calibración. Errores en \mathbf{K} o sesgos en Z se traducen en distorsiones geométricas que afectan a etapas posteriores como el registro y la evaluación métrica. [13], [14]

2.4. Preprocesamiento de nubes de puntos

El preprocesamiento limpia y estructura la nube antes de compararla con el modelo teórico. En impresión por capas se trabaja a una corta distancia del objeto, así que pequeñas variaciones del sensor conllevan a un error. En caracterizaciones recientes de las familias RealSense D400 y Azure Kinect se observa que el error sistemático y la resolución efectiva cambian con el modo y la distancia, lo que obliga a fijar rangos de trabajo y parámetros de filtrado acordes a cada dispositivo. Esta evidencia empírica sirve para decidir un recorte por rango, como limitar z a la ventana útil de la impresora y descartar puntos fuera del volumen donde realmente se deposita material. [19]

Separar la pieza del entorno es el siguiente paso práctico. Una estrategia muy estable, y barata computacionalmente, consiste en detectar y sustraer planos dominantes como la cama de impresión mediante ajuste robusto con RANSAC, de esta forma se segmentan los inliers del plano para estimar su ecuación y con los outliers como candidatos de la pieza. La biblioteca PCL ofrece este flujo y lo documenta con ejemplos reproducibles donde se fija un umbral de distancia al plano. [17]

Sobre la nube residual se aplica eliminación de ruido. El filtro estadístico por vecindad de PCL estima media y desviación de la distancia a los vecinos y conserva puntos con distancia compatible, lo que reduce salpicaduras y lecturas erráticas sin deformar bordes. En la práctica, conviene ajustar k en función de la densidad tras el muestreo y del tamaño mínimo de detalle que se quiere preservar. [17]

Para controlar la complejidad sin perder geometría relevante se usa Voxel Grid Downsampling. La idea es discretizar el volumen en celdas cúbicas y representar cada celda por un punto, normalmente el centroide. En datos densos, la segmentación basada en rejilla de vóxeles con umbrales de densidad mínimos resulta simple y escalable, y permite regular la conectividad cambiando el tamaño del vóxel. Esta aproximación se ha validado en segmentación de nubes densas donde incluso se extrae primero el suelo y luego se agrupa por adyacencia de vóxeles, un proceso similar a separar cama y pieza. [23]

La elección del tamaño de vóxel no es arbitraria. Debe respetar la curvatura de la pieza y el paso de capa, y a la vez promediar ruido. Un criterio práctico es fijarlo en el orden del error de profundidad cercano del sensor y nunca por debajo de la mitad del espesor de capa, de modo que el muestreo no destruya estructura de capa ni aristas pequeñas. La caracterización comparativa de D 415, D 435i, D 455 y Azure Kinect ofrece rangos y sesgos útiles para esa elección inicial. [19]

Con la nube ya reducida se estiman normales. Hacerlo después del muestreo mejora la estabilidad porque el vecindario es más uniforme. En PCL el proceso típico calcula normales con un radio de búsqueda que suele ser entre dos y tres veces el tamaño del

vóxel, y estas normales alimentan descriptores o métricas posteriores. La propia PCL ilustra este encadenado de estimación de normales y cómputo de descriptores como FPFH dentro de un mismo grafo de procesos; cuando la comparación con el modelo requiera correspondencias apoyadas en geometría local, los FPFH son una opción consolidada por su coste moderado y su capacidad de discriminar en presencia de ruido. [17], [24]

2.5. Registro de nubes de puntos y alineación con el modelo teórico

En el modelado por RGB-D, Prankl y colaboradores presentan un sistema flexible que reconstruye modelos 3D completos a partir de capturas parciales en distintas sesiones y los fusiona de forma automática, de modo que la malla final es métrica y usable para reconocimiento de instancias y seguimiento con pose de seis grados. El proceso no impone restricciones fuertes de configuración, funciona con cámara móvil o plato giratorio, y conecta de forma natural módulos de seguimiento y de reconstrucción. Esta visión de acumular geometría parcial fiable para usarla luego en reconocimiento inspira nuestro enfoque, donde el objetivo no es crear el modelo, sino alinear la nube adquirida con el modelo teórico para medir progreso de fabricación con criterios geométricos claros. [25]

El objetivo del registro es encontrar la transformación rígida que alinea la nube de puntos adquirida en la capa actual con el modelo teórico de la pieza. En términos prácticos se busca calcular una rotación y traslación que minimicen las diferencias geométricas a pesar del ruido, oclusiones y crecimiento parcial de la geometría a lo largo del tiempo. El punto de partida clásico es ICP, que itera entre emparejar puntos cercanos y ajustar una transformación de mínimos cuadrados. Su eficacia es alta cuando existe una buena inicialización, pero puede atascarse en mínimos locales y es sensible a emparejamientos incorrectos y solapes parciales, algo común en impresión por capas. [26], [27]

Por otro lado, también se encuentra TEASER y su implementación acelerada TEASER++ que replantean el registro para que los errores muy grandes casi no influyan en el cálculo. Separan la estimación en pasos simples, primero la escala, luego la rotación y al final la traslación, estos métodos ofrecen garantías de que el resultado es válido. En evaluaciones públicas logran alineaciones correctas con alto porcentaje de acierto y con tiempos aptos para uso en robótica. Estos algoritmos son útiles cuando las capas tempranas contienen poco contexto geométrico. [28]

Go ICP explora el espacio de rotaciones y traslaciones con cotas geométricas y converge al mínimo global sin suposiciones fuertes sobre la inicialización. Su coste crece con el tamaño de las nubes y no siempre es adecuado en línea, pero es una referencia útil para validar resultados o para calibraciones fuera de línea donde el tiempo no es crítico. [29]

En esta investigación se adopta el algoritmo CliReg debido a su capacidad para manejar escenarios con una alta tasa de correspondencias espurias (outliers). La innovación principal del método reside en reformular el problema de consenso máximo como una búsqueda eficiente de cliques maximales en un grafo de correspondencias. A diferencia de enfoques previos que buscan únicamente el clique máximo, CliReg emplea un esquema Branch-and-Bound para enumerar candidatos prometedores y validarlos mediante una función de aptitud (fitness) basada en mínimos cuadrados. Esta estrategia permite aislar transformaciones rígidas coherentes incluso cuando la geometría contiene simetrías o ruido que confundirían a algoritmos tradicionales como RANSAC o TEASER++. [30]

2.6. Monitoreo y control superficial en fabricación aditiva a partir de datos 3D

En los últimos años se han desarrollado distintos procedimientos destinados a monitorear y evaluar el proceso de fabricación aditiva a partir de datos tridimensionales. En primer lugar, existe un enfoque que trabaja directamente con datos 3D para detectar cambios sutiles capa a capa. El pipeline empieza con el escaneo de la pieza, registra la nube actual con las anteriores, divide la superficie en regiones y extrae descriptores locales usando kernel correlation. Estos descriptores alimentan un modelo de cascada profunda y, en paralelo, se usa un gráfico de control en tiempo real para lanzar alertas cuando aparecen desviaciones. En pruebas con fabricación por filamento fundido, el método detectó variaciones pequeñas de temperatura, caudal y altura de capa con alta precisión y logró identificar el cambio a partir de la capa dieciséis, manteniendo tiempos de cómputo aptos para monitoreo in situ. [31]

Otro antecedente es una variante que transforma la nube de puntos a imágenes estructuradas para facilitar el uso de clasificadores profundos. Primero rasterizan la nube hasta obtener una imagen o un mapa de alturas. Después entrenan un modelo, ya sea una cascada profunda o una CNN, y comparan cada capa con una referencia para detectar anomalías. La idea es práctica y sencilla de implementar, pero al aplanar la geometría se pueden perder señales débiles asociadas a cambios globales en la textura 3D. Por eso, trabajos posteriores proponen analizar la nube directamente y extraer descriptores con kernel correlation. Esta línea permite entender con claridad qué se gana y qué se pierde cuando se proyecta el 3D a 2D antes de clasificar. [32]

Otra línea se enfoca en evaluar la calidad superficial combinando geometría 3D y textura visual. Para este flujo se emplea un sensor de luz estructurada de línea, se calibra la cámara con el método de Zhang, se calibra el plano de luz y se realiza la calibración mano ojo con el robot. Con estas bases se reconstruye el perfil 3D, se filtra para reducir

ruido y se calcula un índice de textura que estima la rugosidad de forma inmediata sobre la nube. En pruebas con patrones de referencia, el error de reconstrucción fue cercano a 0,07 milímetros y cumple requisitos de hasta 0,2 milímetros, suficiente para control de superficie en procesos de fabricación aditiva. [33]

Complementario a lo anterior, se ha planteado un enfoque de correlación para asegurar la consistencia metrológica cuando se usan distintos sistemas ópticos. El flujo consiste en medir la misma zona con FVM y SLS, alinear ambas nubes con ICP, extraer secciones transversales equivalentes, calcular parámetros de rugosidad y aplicar contrastes estadísticos y coeficientes de correlación para decidir si las mediciones son equivalentes. El hallazgo clave es que, pese a diferencias notables en muestreo y densidad de nubes, las estimaciones de rugosidad no difieren significativamente entre sistemas cuando se realiza una alineación y selección de secciones rigurosas, lo que aporta confianza para integrar sensores heterogéneos en línea. [34]

Finalmente, existe una propuesta específica de monitorizado superficial con procesamiento in-situ de nubes de puntos durante la fabricación. La idea central es capturar y procesar la geometría 3D de la superficie conforme avanza la impresión, de forma que se puedan identificar desviaciones y defectos sin detener el proceso, apoyándose en cómputo a bordo y análisis geométrico directo de la nube. Este enfoque consolida la viabilidad de pipelines en los que la nube de puntos es el dato primario para decidir sobre el estado de la fabricación. [35]

En conjunto, estos trabajos comparten una estructura de pipeline muy clara. Adquisición 3D capa a capa, registro contra una referencia, reducción o partición espacial con descriptores locales y una etapa de decisión con aprendizaje o estadística.

2.7. Síntesis del estado del arte

El análisis del estado del arte presentado en este capítulo ha permitido identificar un conjunto de técnicas y enfoques para la adquisición, el preprocesamiento, el registro y la evaluación geométrica de nubes de puntos en contextos de fabricación aditiva. A partir de esta revisión, se han extraído una serie de conclusiones que han guiado las decisiones metodológicas adoptadas en el desarrollo del sistema propuesto.

En primer lugar, respecto a la adquisición de información tridimensional, para escenarios de reconstrucción cercana los sensores RGB-D de corto alcance son una mejor opción ya que ofrecen una buena relación entre densidad de puntos, estabilidad temporal y facilidad de integración. Frente a métodos pasivos basados en fotogrametría o SfM, que requieren múltiples vistas y presentan dificultades en superficies de baja textura, y a sensores ToF orientados a rangos mayores pero con suavizado, se ha optado por un sensor

RGB-D de la familia Intel RealSense. Esta elección se apoya por su uso a corta distancia y en la estabilidad de sus parámetros intrínsecos y extrínsecos.

En cuanto a la calibración, el estado del arte muestra que la coherencia entre capturas es un requisito indispensable para cualquier comparación geométrica fiable. Por este motivo, el sistema desarrollado adopta un flujo de calibración explícito que combina la calibración intrínseca de los canales RGB e infrarrojo, la estimación de la transformación extrínseca y el uso de marcadores fiduciales para fijar un marco de referencia estable en la celda de impresión. Esta decisión se alinea con los procedimientos descritos en la literatura y permite garantizar que las diferencias observadas entre nubes consecutivas se deban a cambios reales en la geometría impresa y no a variaciones en la pose del sensor o a errores de alineación.

Respecto al preprocesamiento de nubes de puntos, la revisión evidencia que la calidad del registro posterior depende en gran medida de la limpieza y estructuración previa de los datos. Por ello, se ha optado por un encadenado de operaciones que incluye recorte por rango, eliminación de planos dominantes, filtrado estadístico, filtrado por densidad y voxelización. En particular, el uso de una discretización basada en vóxeles se justifica por su capacidad para reducir la complejidad computacional sin destruir la estructura geométrica relevante, siempre que el tamaño del vóxel se seleccione en relación con el error del sensor y el espesor de capa.

En el ámbito del registro de nubes de puntos, el estado del arte muestra que los métodos clásicos como ICP presentan limitaciones importantes en escenarios con solapamiento parcial, ruido y ausencia de una inicialización cercana. Del mismo modo, los métodos globales basados en RANSAC o en optimización robusta ofrecen mejoras, pero pueden degradarse cuando la proporción de correspondencias erróneas es elevada. En este contexto, se ha adoptado el algoritmo CliReg como método principal de registro, dado que al estar basada en cliques permite identificar subconjuntos de correspondencias geométricamente coherentes incluso en presencia de un alto número de outliers. Esta elección está directamente motivada por las condiciones propias de la impresión 3D por capas, donde la geometría observada es incompleta y evoluciona progresivamente.

Finalmente, en relación con el monitoreo y la evaluación del progreso de impresión, la literatura revisada evidencia dos grandes enfoques. Aquellos basados en proyecciones bidimensionales y aprendizaje profundo, y aquellos que operan directamente sobre la geometría tridimensional. En este trabajo se ha optado por un enfoque puramente geométrico, apoyado en métricas de solapamiento, cobertura superficial y progreso en altura, que permiten una interpretación directa y trazable de los resultados sin necesidad de entrenamiento previo..

En conjunto, el estado del arte no solo ha servido como marco teórico de referencia,

sino que ha permitido justificar de manera coherente cada una de las elecciones realizadas en el diseño del sistema.

Metodología

En este capítulo se describe la metodología seguida para diseñar, implementar y comprobar el sistema propuesto de validación de impresiones 3D basado en nubes de puntos. Su función es establecer cómo se han desarrollado los objetivos planteados, teniendo en cuenta el desarrollo de software, el montaje experimental y la evaluación cuantitativa sobre una pieza de referencia.

3.1. Objetivo y enfoque general

Partiendo de los objetivos específicos, la metodología se estructura en varias fases. En primer lugar, se define la arquitectura del sistema, identificando los módulos funcionales necesarios y las relaciones entre ellos. A continuación, se procede a la implementación modular de estos componentes en Python, de forma que cada bloque pueda desarrollarse, probarse y reutilizarse de manera independiente. Sobre esta base, se diseña y monta una configuración experimental controlada para capturar nubes de puntos durante el proceso de impresión 3D y compararlas con el modelo teórico de la pieza. Finalmente, se plantean múltiples ensayos sobre una pieza de referencia para cuantificar el comportamiento del sistema mediante un conjunto de métricas de alineación y progreso de impresión.

En conjunto, la metodología combina tres etapas. Por un lado, el diseño de una arquitectura software que soporte el flujo completo desde la adquisición de datos hasta la evaluación de resultados. Por otro, la construcción de una configuración física reproducible, en el que se integran la impresora 3D, el sensor RGB-D, los marcadores ArUco y la pieza de ensayo. Por último, la definición de un procedimiento de validación cuantitativo sobre una pieza de referencia. De esta forma se puede analizar de forma sistemática la robustez del sistema frente a diferentes configuraciones de captura y niveles de avance de

la impresión.

3.2. Entorno hardware y configuración experimental

El entorno físico se ha diseñado utilizando una impresora 3D FDM marca Prusa. Esta representa un escenario de fabricación por capas desde la cual se ejecuta el sistema de adquisición y análisis de nubes de puntos.

La captura de la escena se realiza usando una cámara RGB-D montada sobre un brazo articulado con cuatro grados de libertad. El brazo se fija al borde de la mesa mediante una mordaza a 30 cm de la impresora. Con este soporte se genera una sujeción rígida y permite ajustar la posición relativa de la cámara respecto a la impresora. A través de las articulaciones del brazo se puede modificar la altura, el alcance y el ángulo de visión, hasta situar la cámara en una configuración que garantice la observación de la pieza y de los marcadores ArUco. Estos se encuentran ubicados en unos soportes alrededor de la zona de impresión y proporcionan un sistema de referencia claramente identificable que se utiliza para estimar la pose relativa del sensor y normalizar las capturas. En la Figura 3.1 se puede observar todo el montaje físico del sistema.



Figura 3.1 Montaje de los componentes

Para los experimentos se ha utilizado una pieza descargada de la web. Esta se modificó para que esté compuesta por tres bloques, una base cuadrada, un cuerpo intermedio cilíndrico y una sección superior. Cada bloque se ha impreso como elemento independiente y puede ensamblarse usando unas barras de unión. Esta modularidad facilita la repetición de ensayos. En la Figura 3.2 se observa la pieza utilizada y su representación de diferentes niveles.

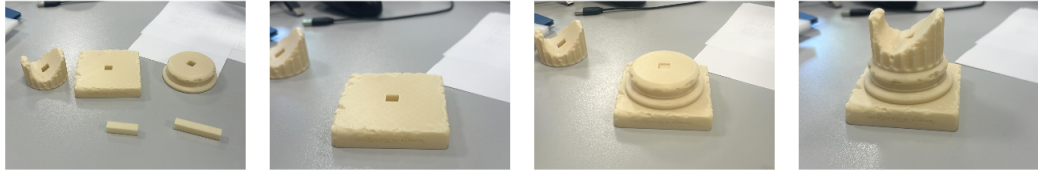


Figura 3.2 Pieza impresa apilable

3.3. Entorno software y herramientas de desarrollo

El sistema propuesto se ha implementado y evaluado en un entorno software orientado a la reproducibilidad y al desarrollo modular. Para ello se ha adoptado una configuración basada en herramientas de uso extendido en visión por computador y procesamiento 3D, de modo que tanto la adquisición con sensores RGB-D como el preprocesamiento, registro y visualización de nubes de puntos puedan ejecutarse en una misma plataforma.

Por otro lado, el desarrollo completo del sistema, incluyendo el código fuente, scripts de ejecución y utilidades auxiliares, se encuentra disponible en un repositorio público. Dicho repositorio recoge la implementación descrita en este capítulo y permite la reproducción de los experimentos presentados en este trabajo. El código puede consultarse en:

<https://github.com/Diegotripi/>

TFM-Validacion-Geometrica-en-Impresion-3D-mediante-Vision-RGB-D

3.3.1. Sistema operativo y configuración base

Todo el desarrollo del sistema se ha realizado sobre un entorno GNU/Linux basado en Ubuntu 22.04.5 LTS (Jammy), que actúa como sistema operativo principal tanto para la adquisición de datos como para el procesamiento de las nubes de puntos. Sobre esta plataforma se ha utilizado Python 3.10 como lenguaje de programación, aprovechando su ecosistema de librerías orientadas a visión por computador, tratamiento geométrico y visualización.

El sistema se ha ejecutado en un único PC, que integra la conexión con la cámara Intel RealSense y aloja el código del proceso. La elección de un entorno estándar como Ubuntu 22.04 facilita la instalación de las dependencias necesarias y favorece la reproducibilidad del trabajo en otras máquinas que dispongan de una configuración similar.

3.3.2. Librerías del software

El núcleo del sistema se apoya en un conjunto de librerías especializadas que dan soporte a las distintas etapas del pipeline. La comunicación con la cámara Intel RealSense se realiza mediante la librería `pyrealsense2`, que proporciona una interfaz de bajo nivel para configurar la cámara, adquirir imágenes sincronizadas de color y profundidad y acceder a sus parámetros intrínsecos.

Las operaciones sobre nubes de puntos se llevan a cabo principalmente con `open3d`, que ofrece estructuras de datos para representar nubes, algoritmos de filtrado y down-sampling, métodos de registro geométrico y herramientas de visualización. Para el cálculo numérico y las operaciones geométricas básicas se utilizan `numpy` y `scipy`, que facilitan la implementación de transformaciones, cálculos de distancias y manipulación de matrices y vectores.

El tratamiento de las imágenes RGB y la detección de marcadores ArUco se realiza con `opencv-python`, que permite cargar, procesar y mostrar imágenes, así como estimar poses de cámara a partir de los marcadores presentes en la escena. Por último, el registro robusto entre la nube generada y el modelo teórico se apoya en la librería `clireg`, que implementa los algoritmos CliReg y CliRegMutual propuestos en la literatura y los expone como funciones utilizables desde Python. Cada una de estas librerías se integra en los módulos correspondientes del pipeline.

3.4. Estructura modular del proyecto

El código desarrollado se organiza siguiendo una estructura modular que refleja las etapas principales del algoritmo de validación. Cada módulo agrupa las funcionalidades asociadas a una fase concreta del proceso y expone una interfaz bien definida hacia el resto del sistema, lo que facilita tanto el desarrollo como la depuración y la futura extensión del trabajo.

El módulo de adquisición de datos agrupa las rutinas relacionadas con la adquisición de datos mediante la cámara Intel RealSense. En él se incluyen las funciones para inicializar el dispositivo, configurar los parámetros de captura y construir nubes de puntos a partir de los pares de imágenes de color y profundidad. El módulo preprocesado contiene los filtros y transformaciones aplicados a las nubes de puntos brutas, las operaciones de recorte espacial y las normalizaciones necesarias para expresar las nubes en un sistema de referencia común.

El registro geométrico entre la nube generada y el modelo teórico se implementa en el módulo de registro, donde se integran distintos métodos de alineación. A partir de las

nubes registradas, el módulo evaluación se encarga del cálculo de las métricas de interés, tales como la altura relativa, la cobertura superficial o la coincidencia volumétrica. Estas métricas se utilizan posteriormente para analizar el progreso de la impresión y la calidad del registro.

La generación de vistas y figuras se centraliza en el módulo de visualización, que proporciona utilidades para representar nubes de puntos en 3D, proyectarlas sobre planos bidimensionales y resaltar regiones de interés. Complementariamente, el módulo de utilidades agrupa funciones de apoyo de propósito general, como la carga de modelos teóricos, la lectura de ficheros de configuración o utilidades geométricas auxiliares.

Esta organización modular responde a una filosofía de separación de responsabilidades, donde cada bloque resuelve una tarea del proceso global. La combinación de todos los módulos permite recorrer de forma estructurada el flujo completo, desde la adquisición de nubes de puntos hasta la obtención de métricas cuantitativas y figuras de visualización, manteniendo al mismo tiempo la legibilidad y mantenibilidad del código.

3.5. Estrategia de desarrollo y validación

La implementación del sistema se ha abordado con una metodología orientada a asegurar consistencia técnica y trazabilidad de resultados. En lugar de desarrollar el sistema como un único bloque, se ha definido una estrategia que combina construcción incremental por módulos y validación progresiva, de forma que cada componente se incorpore al flujo sólo cuando su funcionamiento haya sido verificado. Este enfoque permite controlar la complejidad del sistema, reducir el riesgo de errores y facilitar la repetición de experimentos bajo condiciones comparables.

3.5.1. Desarrollo incremental por módulos

El desarrollo del sistema se ha llevado a cabo siguiendo una estrategia incremental basada en módulos, en la que cada bloque funcional del algoritmo se ha implementado y verificado de manera progresiva. En una primera fase se abordó la implementación del módulo de adquisición, garantizando la correcta comunicación con la cámara RealSense y la construcción de nubes de puntos coherentes a partir de los datos de color y profundidad. Una vez consolidada esta etapa, se desarrollaron los módulos de preprocesado y registro, y finalmente se incorporaron las funcionalidades de evaluación y visualización.

Este enfoque ha permitido ejecutar y depurar cada módulo de forma aislada, lo que facilita la identificación de errores y la comprobación de que cada bloque cumple su función antes de integrarlo en el flujo completo. Un caso son las rutinas de preprocesado,

que se han probado inicialmente sobre nubes almacenadas en disco, sin depender de la adquisición en tiempo real, mientras que los algoritmos de registro se han validado primero sobre pares de nubes controladas antes de aplicarlos directamente a las capturas obtenidas en la configuración física.

La integración gradual de los módulos ha dado lugar a un algoritmo coherente donde cada componente interactúa con el resto a través de interfaces bien definidas. Este procedimiento reduce el riesgo de errores difíciles de localizar y proporciona una base sólida para introducir mejoras o extensiones futuras sin comprometer el funcionamiento global del sistema.

3.5.2. Validación por etapas

La validación del sistema se ha planteado igualmente de forma escalonada, evaluando de manera separada el comportamiento de cada módulo antes de analizar el desempeño global. En el caso del módulo de adquisición, se han verificado la coherencia geométrica de las nubes de puntos generadas y la estabilidad de las capturas frente a pequeñas variaciones en la configuración de la cámara o en la iluminación del entorno.

Sobre las nubes adquiridas, el módulo de preprocesado se ha validado comprobando que los filtros aplicados reducen eficazmente el ruido y los puntos fuera de rango, preservando al mismo tiempo la geometría relevante de la pieza. Para el registro, se ha analizado la capacidad de los métodos implementados para converger hacia alineaciones correctas bajo diferentes condiciones de captura, comparando su comportamiento con técnicas más convencionales como ICP o RANSAC. Esta comparación permite situar el rendimiento de CliReg en un contexto conocido y valorar sus ventajas en términos de robustez frente a correspondencias erróneas.

Finalmente, la validación de las métricas de evaluación se ha orientado a comprobar su coherencia con el progreso teórico de la impresión y con la percepción cualitativa de las nubes registradas. Para ello, se han calculado métricas como *zmax_percent*, *progress_surface* y *perc_b_in_a* en distintos estados de la pieza y se ha verificado que su evolución refleja adecuadamente el avance de la fabricación y las discrepancias entre la geometría teórica y la observada. El detalle de estos experimentos, junto con los resultados numéricos y las figuras asociadas, se presenta en el capítulo de Resultados y discusión; en este capítulo se describe únicamente el enfoque metodológico seguido para estructurar dicha validación.

3.5.3. Relación con el diseño experimental

La metodología descrita se concreta en el capítulo de resultados mediante un conjunto de escenarios experimentales que combinan distintos niveles de progresión de la pieza y

configuraciones de captura. En cada escenario se ejecuta el algoritmo completo, desde la adquisición de las nubes de puntos hasta el cálculo de las métricas de evaluación, utilizando siempre el hardware descrito y el entorno software definido en este capítulo.

El uso de ficheros de configuración y scripts de ejecución reproducibles permite repetir los experimentos bajo las mismas condiciones hardware y software, lo que facilita comparar resultados entre sesiones y analizar el efecto de modificaciones concretas en la configuración del sistema. De este modo, la estrategia de desarrollo incremental y validación por etapas se integra con el diseño experimental planteado, proporcionando un marco sistemático para estudiar el comportamiento del método propuesto y extraer conclusiones sobre su capacidad para validar el progreso de impresiones 3D a partir de nubes de puntos.

Arquitectura del sistema de adquisición y análisis de nubes de puntos

En este capítulo se describe la arquitectura global del sistema de adquisición de nubes de puntos desarrollado para monitorizar el proceso de impresión 3D. Tal y como se resume en la Figura 4.1, el sistema se organiza en módulos que transforman progresivamente la información capturada por el sensor RGB-D hasta obtener métricas de convergencia y visualización. En concreto, la arquitectura se estructura en los siguientes bloques funcionales:

- **Adquisición de datos:** genera nubes de puntos normalizadas a partir de pares sincronizados de color y profundidad. Garantiza que todas las capturas se encuentren en un marco de referencia común.
- **Representación geométrica:** construye el modelo teórico de referencia a partir del fichero `.stl` o de una nube existente. Unifica unidades, posición y discretización espacial.
- **Preprocesamiento de nubes:** filtra ruido, elimina outliers y ajusta la resolución mediante operaciones de voxelización, filtrado estadístico, depuración por densidad y recortes geométricos. Deja la nube de puntos adquirida experimentalmente y el modelo adecuadas para el registro.
- **Alineación:** estima una transformación entre la nube generada y el modelo teórico utilizando métodos de registro y refinamientos locales.
- **Evaluación y comparación:** calcula métricas de error, solapamiento y progreso de la impresión. Cuantificando la calidad del registro y el grado de avance geométrico.

- **Visualización:** genera representaciones bidimensionales, principalmente proyecciones en el plano YZ , que facilitan la inspección cualitativa de la alineación, solapamiento y progresión en altura de la pieza.

En conjunto, la arquitectura mostrada en la Figura 4.1 define un proceso modular en el que cada etapa dispone de responsabilidades y proporciona las entradas necesarias para la siguiente. Además, se garantiza así un flujo de datos consistente desde la captura inicial hasta el análisis final de la impresión.



Figura 4.1 Arquitectura general del sistema de adquisición y análisis de nubes de puntos

4.1. Adquisición de datos

El objetivo del módulo de adquisición de datos es capturar y sincronizar con un sensor RGB-D pares de fotografías de color y profundidad, con la finalidad de construir nubes de puntos de la escena. Posteriormente, se filtran para segmentar la pieza de interés. El desempeño de las siguientes fases depende de la calidad y la consistencia de las nubes generadas.

4.1.1. Normalización geométrica de la escena

Aunque la nube de puntos se obtiene en el marco de la cámara, su utilidad para comparación geométrica requiere expresarla en un sistema de referencia fijo y repetible entre capturas. En el entorno de impresión, la cámara puede variar su posición u orientación, por lo que sin una referencia externa las nubes adquiridas no serían directamente comparables. Para resolverlo, se emplean marcadores ArUco como referencia espacial, lo que permite estimar la pose relativa del sensor respecto a la región de impresión y transformar la nube a un marco común ligado a la celda. Este paso actúa como enlace entre la adquisición y las etapas posteriores de filtrado, segmentación y registro, al garantizar consistencia geométrica entre sesiones. [36]

La normalización de la nube de puntos se logra mediante la construcción de una transformación rígida definida por una matriz de rotación R y un vector de traslación t . Estos parámetros describen la posición del plano que contiene los marcadores ArUco y su orientación respecto al sistema de cámara. [36]

Para ello, primero se estiman los centros tridimensionales de los marcadores a partir de la proyección inversa de sus esquinas detectadas en la imagen RGB. Posteriormente, se define un plano a partir de los marcadores visibles calculando su centroide y su vector normal mediante una descomposición en valores singulares, lo que garantiza una estimación frente a pequeñas perturbaciones en la medida.

Una vez obtenidas las coordenadas de los marcadores ArUco en referencia a la cámara, es posible estimar el plano que los contiene y, a partir de él, definir el origen y la orientación del nuevo marco de referencia. Sea el conjunto de marcadores detectados $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4\}$, con $\mathbf{p}_i = [x_i, y_i, z_i]^T$ las posiciones de los centros de los marcadores expresadas en coordenadas de cámara.

El centroide del conjunto de puntos se define como la media de sus coordenadas:

$$\mathbf{c} = \frac{1}{N} \sum_{i=1}^N \mathbf{p}_i, \quad (4.1)$$

Donde N es el número de marcadores visibles, con la restricción de que $N \geq 3$

Una vez calculado el centroide, se obtiene la matriz de desviaciones que describe cómo se distribuyen los puntos alrededor del centroide:

$$\mathbf{A} = \begin{bmatrix} (\mathbf{p}_1 - \mathbf{c})^T \\ (\mathbf{p}_2 - \mathbf{c})^T \\ \vdots \\ (\mathbf{p}_N - \mathbf{c})^T \end{bmatrix}, \quad (4.2)$$

Para estimar la orientación del plano que mejor se ajusta al conjunto de puntos se aplica la descomposición en valores singulares a la matriz \mathbf{A} :

$$\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T, \quad (4.3)$$

Donde \mathbf{U} y \mathbf{V} son matrices ortogonales, y $\mathbf{\Sigma}$ es una matriz diagonal que contiene los valores singulares ordenados de mayor a menor. Cada valor singular en $\mathbf{\Sigma}$ indica la magnitud de la varianza del conjunto de puntos a lo largo de las direcciones principales descritas por los vectores de \mathbf{V} . En este contexto, los dos valores singulares mayores corresponden a las direcciones del plano que mejor se ajusta a los puntos, mientras que el valor singular más pequeño representa la dirección con menor varianza, es decir, la normal al plano.

El vector normal unitario al plano se identifica como el último vector singular de \mathbf{V} , correspondiente al menor valor singular:

$$\mathbf{n} = \mathbf{v}_3, \quad (4.4)$$

Con \mathbf{v}_3 siendo la tercera columna de \mathbf{V} . Este vector define la dirección perpendicular al plano que mejor se ajusta a los puntos observados en un sentido de mínimos cuadrados.

De este modo, la ecuación del plano que contiene los marcadores puede escribirse como:

$$\mathbf{n}^T(\mathbf{x} - \mathbf{c}) = 0, \quad (4.5)$$

Donde \mathbf{x} representa un punto cualquiera del plano, \mathbf{n} es el vector normal unitario y \mathbf{c} es el centroide.

Este procedimiento tiene la ventaja de ser robusto frente al ruido de medida y a pequeñas inexactitudes en la localización de los marcadores, ya que busca la orientación del plano que minimiza el error cuadrático total. En el contexto del sistema, \mathbf{c} se adopta como el origen del marco de referencia local asociado a la superficie de impresión, mientras que el vector normal \mathbf{n} proporciona la dirección del eje \mathbf{Z} del nuevo sistema de coordenadas.

A partir de esta estimación, se definen los ejes \mathbf{X} y \mathbf{Y} mediante relaciones vectoriales entre los marcadores. En concreto, se define el eje \mathbf{X} con la misma dirección y sentido que el vector que se forma entre el marcador 2 y 1. Por otro lado, se realiza el mismo procedimiento con el eje \mathbf{Y} entre los marcadores 3 y 2; además, se comprueba que este es ortogonal al eje \mathbf{X} . Finalmente, la matriz de rotación \mathbf{R} resultante se compone concatenando estos tres vectores unitarios como filas:

$$\mathbf{R} = \begin{bmatrix} \mathbf{X}^T \\ \mathbf{Y}^T \\ \mathbf{Z}^T \end{bmatrix}, \quad (4.6)$$

Que junto con la traslación $\mathbf{t} = \mathbf{c}$, define la transformación rígida empleada para alinear la nube de puntos al marco de referencia normalizado según:

$$\mathbf{X}' = \mathbf{R}(\mathbf{X} - \mathbf{t}). \quad (4.7)$$

Esta transformación permite expresar todas las nubes de puntos en un sistema común, independiente de la posición o el ángulo de la cámara, garantizando así una base geométrica consistente para las etapas de filtrado, registro y comparación.

4.1.2. Delimitación de la región de interés

Una vez que la nube ha sido normalizada, el siguiente paso consiste en delimitar la región de interés.

Este recorte se realiza mediante una caja alineada y centrada en el origen, cuyos planos son paralelos a los ejes del sistema. En cuanto a las dimensiones de la caja, la longitud de esta en el eje **X** y el eje **Y** corresponde al largo y ancho de la cama de impresión. Por otro lado, la altura de esta corresponde a la altura del marcador ArUco que se encuentra en el extrusor.

Finalmente, con este recorte se reduce considerablemente el número de puntos, lo que acelera el preprocesamiento, registro y la comparación, con menores requisitos de memoria.

4.1.3. Arquitectura software del módulo

El módulo de adquisición de datos se organiza en cinco componentes principales: Inicialización y configuración del sensor RGB-D, Adquisición sincronizada de color y profundidad, Construcción de la nube de puntos, Normalización del espacio y Recorte de la región de interés.

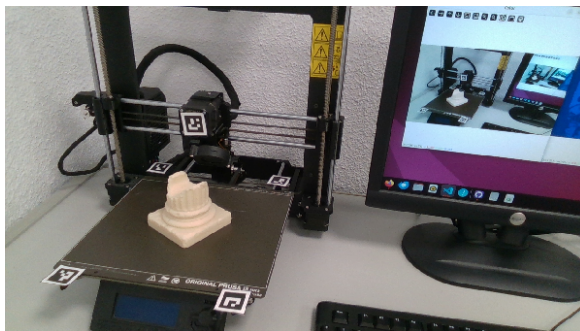
La figura A.1 resume las responsabilidades de cada clase y sus relaciones. **Realsense** encapsula la inicialización del sensor RGB-D, la configuración del sensor y la obtención sincronizada de color y profundidad. **utils_projection** implementa la proyección inversa y la estimación de la profundidad local, conectando directamente con la formulación de (2.1) y (2.2). **BackgroundCapturePipeline** maneja la detección de los marcadores ArUco, el cálculo de centros 3D, el ajuste del plano y la construcción del marco ortonormal, aplicando después la transformación de rotación y traslación (4.7) para obtener una nube normalizada. Finalmente, **CropBox** formaliza el recorte para delimitar la zona de interés.

Flujo de ejecución

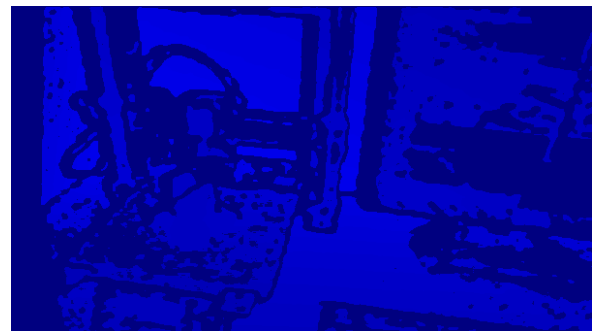
1. **Inicialización y configuración del sensor RGB-D:** La inicialización parte de un archivo Notación de Objetos JavaScript (JSON) y la función `setup_camera`. Allí se especifica la resolución y el número de cuadros por segundo que puede capturar el sensor. Por defecto se define una resolución estándar de 1280×720 y 30 cuadros por segundo. Adicionalmente, se define un umbral de recorte, usado para descartar píxeles a cierta distancia de la cámara; este se configura por defecto a 20 cm. Finalmente, la clase **Realsense** da la opción de calentar el sensor previo a realizar las capturas y configurar una serie de filtros que se encuentran en el entorno de

programación del Kit de Desarrollo de Software (SDK):

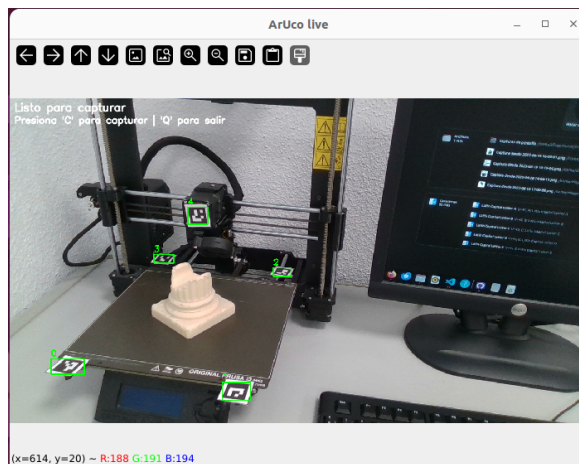
- **Suavizado:** atenúa ruido de alta frecuencia y píxeles dispersos en el mapa de profundidad preservando discontinuidades en bordes geométricos. Suele combinar un suavizado bilateral/guía (sobre la propia profundidad o la imagen de color) con submuestreo espacial para reducir granulado y coste computacional.
 - **Filtro temporal:** estabiliza fluctuaciones cuadro a cuadro modelando la profundidad como un proceso suavizado en el tiempo. Una forma típica es un filtro recursivo de primer orden:
 - **Relleno de huecos:** interpola zonas sin medida por oclusiones o baja reflectancia. Utiliza vecindades locales para cerrar agujeros pequeños.
 - **Fusión temporal por mediana:** acumula N mapas de profundidad alineados y calcula la mediana por píxel
2. **Adquisición sincronizada de color y profundidad:** esta etapa garantiza que cada imagen RGB tenga su correspondiente mapa de profundidad, de modo que exista correspondencia píxel a píxel antes de la reconstrucción (2.2). Operativamente, el flujo arranca solicitando a la cámara un par de capturas, color `bgr8` y profundidad `z16`, y descartando cualquier captura que llegue sin marca temporal válida o con tamaños dispares. A continuación, se aplica el alineamiento mediante `get_aligned_frames`, preservando la geometría interna del modelo pinhole (2.1). Con el par alineado, se realiza la conversión de unidades y el recorte por distancia para eliminar píxeles fuera del rango útil de la escena. Si se ha habilitado la fusión temporal por mediana, la adquisición realiza N capturas. En caso de desincronización, el módulo descarta el par y solicita una nueva captura para mantener la coherencia temporal. Este control de calidad previo es importante, ya que asegura la consistencia de los datos registrados. En las Figuras 4.2a y 4.2b se puede observar una demostración de capturas. El sistema ofrece dos modos de operación. En modo interactivo se presenta una vista previa con superposición de ArUcos, Figura 4.2c, y captura bajo demanda; en modo automático la captura se dispara cuando los ArUco requeridos aparecen estables durante un número prefijado de cuadros consecutivos.



(a) Captura de color



(b) Captura de profundidad



(c) Vista previa

Figura 4.2 Ejemplo de capturas del sistema durante la adquisición.

3. **Construcción de la nube de puntos:** a partir del par de capturas alineadas $(I_{\text{rgb}}, D_{\text{aligned}})$, se reconstruye la geometría 3D aplicando la proyección inversa (2.2) con los intrínsecos \mathbf{K} de (2.1) de la cámara. Para cada píxel válido (u, v) con profundidad Z , se calcula el punto en el sistema de cámara, obteniéndose un conjunto $\mathcal{P} = \{(X_i, Y_i, Z_i)\}_{i=1}^N$. En esta etapa se descartan píxeles sin medida, saturados o fuera de rango, y se preserva la correspondencia asignando a cada punto su color (R, G, B) proveniente de I_{rgb} , generando una nube coloreada $\{(X_i, Y_i, Z_i, R_i, G_i, B_i)\}$. El resultado de esta etapa es una nube en el marco de la cámara, lista para su normalización espacial en la sección siguiente, Figura 4.3.



Figura 4.3 Nube de puntos generada de las capturas 4.2a y 4.2b

4. **Normalización del espacio:** una vez construida la nube en el sistema de cámara, se lleva a un marco de referencia común definido por los marcadores ArUco. Primero, se detectan los marcadores en I_{rgb} y se obtienen sus centros 3D por proyección inversa 2.1 y 2.2. Con el conjunto $\{\mathbf{p}_i\}_{i=1}^N$ se calcula el centroide \mathbf{c} 4.1 y la matriz de desviaciones \mathbf{A} 4.2. A partir de \mathbf{n} y de direcciones definidas por pares de marcadores, se construyen los ejes $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ ortonormales; se compone la matriz de rotación \mathbf{R} 4.6 y se fija la traslación $\mathbf{t} = \mathbf{c}$. Finalmente, cada punto \mathbf{X} de la nube se transforma como 4.7, obteniendo una representación coherente e invariante a la pose de la cámara. Todo este proceso se realiza en `BackgroundCapturePipeline` usando funciones de `utils.projection`.

Una vez definida la transformación de alineamiento mediante los marcadores, se registra la pose de la cámara en el marco normalizado. El origen de la cámara en el

marco alineado es

$$\mathbf{C} = -\mathbf{R} \mathbf{t} \quad (4.8)$$

y los ejes unitarios de la cámara expresados en el mismo marco se obtienen aplicando \mathbf{R} a los ejes canónicos de la cámara:

$$\text{forward} = \mathbf{R} [0, 0, 1]^\top, \quad \text{up} = \mathbf{R} [0, 1, 0]^\top, \quad \text{right} = \mathbf{R} [1, 0, 0]^\top \quad (4.9)$$

Este cálculo se implementa en la clase `BackgroundCapture` mediante las funciones `_camera_pose_dict` y `_camera_pose_json`, que construyen y escriben un diccionario con \mathbf{C} y los tres vectores unitarios en el marco alineado. `_capture_once` realiza esta escritura junto a la exportación del Formato de Archivo Polygon (PLY) cuando el `output` es a disco, o conserva ambos en memoria (`self.last_pcd`, `self.last_pose_dict`) cuando se selecciona el modo en memoria. A continuación se presenta el formato del diccionario que guarda la posición de la cámara.

Esquema del JSON de pose. El archivo contiene cuatro campos, todos en *metros* y expresados en el marco alineado:

- `camera_position` $\{x, y, z\}$ → posición 4.8;
- `camera_forward_unit` $\{x, y, z\}$ → vector *forward* unitario;
- `basis_right` $\{x, y, z\}$ → vector *right* unitario;
- `basis_up` $\{x, y, z\}$ → vector *up* unitario.

5. **Recorte de la región de interés:** tras la normalización la nube queda expresada en un marco común que permite aplicar un recorte geométrico consistente. Se define una caja rectangular centrada en el origen y dimensiones (L_x, L_y, L_z) acordes al área útil de impresión: L_x y L_y se ajustan al largo y ancho de la cama, mientras que L_z se fija en función de la altura del extrusor. El recorte se implementa evaluando la pertenencia de cada punto transformado $\mathbf{X}' = (x', y', z')$ al paralelepípedo:

$$-\frac{L_x}{2} \leq x' \leq \frac{L_x}{2}, \quad -\frac{L_y}{2} \leq y' \leq \frac{L_y}{2}, \quad -0,02 \leq z' \leq L_z.$$

Los puntos que no satisfacen estas desigualdades se descartan. Este procedimiento, reduce la cantidad de datos y acelera las etapas posteriores de manteniendo únicamente la geometría relevante para el análisis. En la Figura 4.4 se presenta una nube de puntos resultante del proceso descrito.

A continuación se resume en la Tabla 4.1 la función que cumple cada clase para las etapas descritas. Esta tabla complementa el diagrama Lenguaje Unificado de Modelado (UML) A.1, permitiendo identificar las funcionalidades de cada clase.

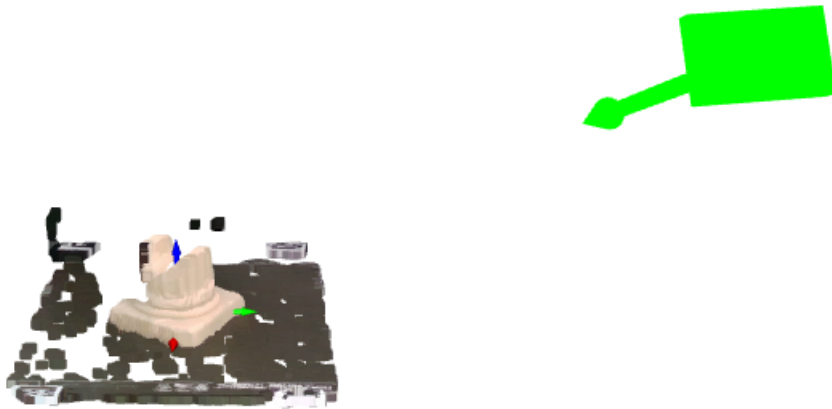


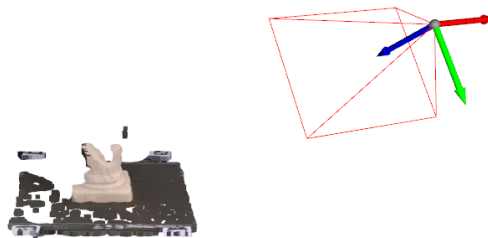
Figura 4.4 Resultado final de la adquisición

Tabla 4.1 Clases del módulo de adquisición: etapa, responsabilidades y ecuaciones.

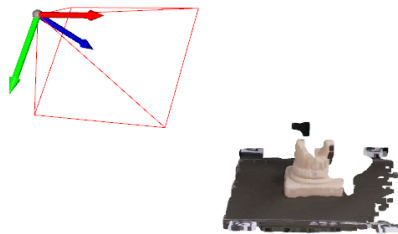
Clase	Etapa	Responsabilidades principales	Ecuaciones
Realsense	Inicialización del sensor RGB-D	Configurar sensor RGB-D, gestionar <code>depth_scale</code> y <code>clip_distance</code> , filtros (suavizado, temporal, relleno de huecos), capturar pares color-profundidad e intrínsecos (f_x, f_y, c_x, c_y) .	(2.1), (2.2).
realsense_utils	Configuración del sensor RGB-D y adquisición sincronizada de color y profundidad	Arrancar proceso de adquisición y alinear profundidad a color (<code>get_aligned_frames</code>); validaciones de tamaño y marca de tiempo.	Previo a (2.2).
utils_projection	Reconstrucción 3D	Proyección inversa y utilidades de profundidad local a partir de (u, v, Z) y \mathbf{K} .	(2.1), (2.2).
BackgroundCapture	Construcción de la nube de puntos y normalización del espacio	Detección/uso de ArUco, centros 3D, marco ortonormal y transformación rígida (\mathbf{R}, \mathbf{t}) .	(4.1), (4.2), (4.3), (4.4), (4.6), (4.7).
Arucodetector	Soporte a normalización	Detección de esquinas/IDs en I_{rgb} ; observaciones para $\{\mathbf{p}_i\}$.	(4.1)–(4.7).
CropBox	Recorte de la región de interés	Caja centrada en el origen con (L_x, L_y, L_z) ; descartar puntos fuera de rango.	Tras (4.7).

4.1.4. Fusión de nubes de puntos

Partiendo de un espacio ya normalizado, la fusión se realiza sumando múltiples nubes parciales capturadas desde distintas poses para obtener una reconstrucción más completa de la región de interés. En la Figura 4.5 se visualiza la fusión de dos nubes obtenidas desde poses diferentes. Este procedimiento, no obstante, tiende a incrementar el error acumulado debido a imprecisiones en las poses, ruido de profundidad y los solapes desalineados que introducen duplicidades y dispersiones. Es responsabilidad de etapas posteriores limpiar la nube fusionada y reducir dicho error antes del análisis métrico.



(a) Nube de la pose 1



(b) Nube de la pose 2



(c) Fusión

Figura 4.5 Fusión de nubes de puntos a partir de dos poses distintas de la cámara.

4.2. Representación geométrica

El módulo de representación geométrica tiene como objetivo unificar, normalizar y discretizar la información de la adquisición para su análisis. En particular, este módulo se encarga de construir una nube de puntos de referencia a partir del modelo teórico y generado de la pieza, y de proporcionar herramientas para proyectar las nubes de puntos sobre diferentes planos de observación.

De este modo, el modelo teórico y las nubes obtenidas experimentalmente pueden describirse en un mismo espacio, centrado y escalado. Esta representación es esencial para las etapas de registro y comparación, donde el sistema debe evaluar las diferencias entre la geometría ideal y la geometría observada durante el proceso de impresión.

Además, la proyección de las nubes de puntos sobre planos 2D, XY, XZ o YZ, permite generar vistas simplificadas que facilitan la inspección visual y el análisis cuantitativo de la superficie, la distribución de puntos y la presencia de zonas vacías. En conjunto, la representación geométrica actúa como puente entre la descripción tridimensional del entorno y las herramientas de análisis y validación utilizadas en el resto del sistema.

4.2.1. Fundamentos teóricos

Desde un punto de vista geométrico, la pieza a imprimir puede describirse inicialmente mediante una malla triangular en \mathbb{R}^3 , formada por un conjunto de vértices y caras:

$$\mathcal{M} = (\mathcal{V}, \mathcal{F}), \quad \mathcal{V} = \{\mathbf{v}_i \in \mathbb{R}^3\}_{i=1}^{N_v} \quad \mathcal{F} \subset \{1, \dots, N_v\}^3, \quad (4.10)$$

Donde \mathcal{V} recoge las coordenadas de los vértices y \mathcal{F} define la conectividad triangular. Aunque esta representación es adecuada para el diseño CAD y la simulación, resulta más conveniente trabajar con una nube de puntos uniforme para las tareas de registro y comparación. Por ello, se construye un conjunto de muestras puntuales sobre la superficie de la malla:

$$\mathcal{P} = \{\mathbf{x}_k \in \mathbb{R}^3\}_{k=1}^N \quad (4.11)$$

Donde los puntos \mathbf{x}_k se obtienen mediante un muestreo aproximadamente uniforme sobre las caras de \mathcal{M} . Esta nube de puntos representa la geometría ideal de la pieza con una densidad controlada por el número de muestras N .

En muchos casos, los modelos CAD se expresan en milímetros, mientras que el sistema de adquisición y el resto del pipeline operan en metros. Para garantizar la coherencia de unidades, es necesario aplicar una transformación de escala global. Si \mathbf{x}^{mm} denota un

punto expresado en milímetros, la conversión a metros se realiza mediante:

$$\mathbf{x}^m = s \mathbf{x}^{mm} \quad \text{con} \quad s = 10^{-3}, \quad (4.12)$$

De modo que todas las coordenadas quedan unificadas en el mismo sistema de unidades que utiliza la cámara RGB-D y el resto del sistema.

Una vez fijadas las unidades, es conveniente normalizar la posición de la geometría para facilitar la comparación con las nubes adquiridas. Para ello, se considera la caja delimitadora alineada a los ejes asociada a la nube:

$$\mathbf{b}_{\text{mín}} = \begin{bmatrix} x_{\text{mín}} \\ y_{\text{mín}} \\ z_{\text{mín}} \end{bmatrix}, \quad \mathbf{b}_{\text{máx}} = \begin{bmatrix} x_{\text{máx}} \\ y_{\text{máx}} \\ z_{\text{máx}} \end{bmatrix} \quad (4.13)$$

Y se traslada la geometría de forma que la base de la pieza descansa sobre el plano $z = 0$. Este desplazamiento se consigue aplicando una traslación vertical:

$$\mathbf{x}' = \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ -z_{\text{mín}} \end{bmatrix} \quad (4.14)$$

De modo que la cota mínima pase a ser $z'_{\text{mín}} = 0$. A continuación, se centra la nube en el plano XY haciendo coincidir el centro de la caja con el origen:

$$\mathbf{c}_{xy} = \begin{bmatrix} \frac{x_{\text{mín}} + x_{\text{máx}}}{2} \\ \frac{y_{\text{mín}} + y_{\text{máx}}}{2} \\ 0 \end{bmatrix}, \quad \mathbf{x}'' = \mathbf{x}' - \mathbf{c}_{xy}. \quad (4.15)$$

La combinación de (4.14) y (4.15) define una transformación rígida que sitúa la pieza con su base apoyada en el plano de referencia y centrada en torno al origen del sistema de coordenadas.

Además de la normalización en posición, es frecuente aplicar una discretización espacial mediante la voxelización. El espacio se particiona en celdas cúbicas de lado Δ y cada voxel se asocia a un subconjunto de puntos:

$$\mathcal{V}_j = \{\mathbf{x}''_k \in \mathcal{P} \mid \mathbf{x}''_k \text{ pertenece al voxel } j\}, \quad (4.16)$$

de forma que la nube puede representar cada celda mediante el centroide de sus puntos:

$$\mathbf{q}_j = \frac{1}{|\mathcal{V}_j|} \sum_{\mathbf{x}''_k \in \mathcal{V}_j} \mathbf{x}''_k \quad (4.17)$$

obteniendo una nube de puntos reducida

$$\mathcal{Q} = \{\mathbf{q}_j\}_{j=1}^{N_v}, \quad (4.18)$$

Con un número menor de muestras pero preservando la estructura geométrica global. Este proceso de voxelización permite disminuir la complejidad computacional de las etapas de registro y comparación, manteniendo una aproximación fiel a la superficie original. [17]

Por otro lado, la representación geométrica incluye la proyección ortogonal de la nube de puntos sobre distintos planos coordenados. Dado un punto $\mathbf{x} = (X, Y, Z)^\top$ en el espacio tridimensional, las proyecciones sobre los planos XY , XZ e YZ se definen como:

$$\pi_{XY}(\mathbf{x}) = \begin{bmatrix} X \\ Y \end{bmatrix}, \quad \pi_{XZ}(\mathbf{x}) = \begin{bmatrix} X \\ Z \end{bmatrix}, \quad \pi_{YZ}(\mathbf{x}) = \begin{bmatrix} Y \\ Z \end{bmatrix} \quad (4.19)$$

Aplicando estas proyecciones a todos los puntos de la nube, se obtienen conjuntos de dos dimensiones \mathcal{P}_{XY} , \mathcal{P}_{XZ} y \mathcal{P}_{YZ} que constituyen vistas ortogonales de la geometría.

Esta representación permite analizar la distribución espacial de la nube en cada plano, facilitando la detección de zonas de alta densidad, huecos o irregularidades en la superficie.

4.2.2. Arquitectura software del módulo

El módulo de representación geométrica se estructura en dos componentes principales, cada uno especializado en una funcionalidad distinta pero complementaria: la construcción y normalización de la nube de puntos teórica, y la proyección bidimensional de nubes tridimensionales. La Figura A.2 muestra la arquitectura UML del módulo y resume las clases involucradas y sus responsabilidades.

La clase `HybridPointCloud` encapsula toda la lógica necesaria para transformar el modelo geométrico de referencia en una nube de puntos uniforme y normalizada. Dependiendo del tipo de entrada, la clase carga un fichero `.stl` o recibe directamente una nube de puntos en formato `Open3D`. En el caso de emplear un modelo STL, se aplican tres operaciones fundamentales: conversión de unidades mediante el factor de escala definido en (4.12), normalización de la posición para alinear la base con el plano $z = 0$ y centrar la geometría en el plano XY siguiendo (4.14) y (4.15), y muestreo uniforme sobre la malla para obtener un conjunto de puntos según (4.11). Posteriormente, la nube resultante puede ser discretizada mediante una rejilla de vóxeles, definida por (4.16) y (4.18), generando así una representación más ligera y apta para las etapas posteriores de registro y comparación.

Por su parte, la clase `PointCloudProjector` proporciona las herramientas necesarias

para proyectar la nube tridimensional sobre planos ortogonales y generar imágenes rasterizadas. La proyección se basa directamente en las transformaciones definidas en (4.19), permitiendo obtener vistas bidimensionales de la geometría en los planos XY , XZ e YZ .

En conjunto, ambas clases ofrecen una abstracción clara y modular para gestionar la geometría teórica y sus representaciones derivadas. Mientras que `HybridPointCloud` gestiona el modelo 3D y su normalización en el espacio, `PointCloudProjector` se centra en las vistas proyectadas, permitiendo que el módulo proporcione tanto información tridimensional como representaciones bidimensionales equivalentes, útiles para diagnóstico y análisis métrico.

Flujo de ejecución

El flujo de ejecución de `HybridPointCloud` se organiza de manera secuencial y puede describirse mediante los siguientes pasos:

1. **Carga del modelo geométrico:** El módulo recibe como entrada un fichero `.stl` o una nube de puntos existente. En el primer caso, se carga la malla y, si procede, se aplica la conversión de unidades conforme a la transformación definida en (4.12). En el segundo caso, la nube de puntos se almacena directamente como entrada válida.
2. **Normalización geométrica:** A partir de la geometría cargada, se calcula su caja delimitadora alineada a los ejes y se aplica la traslación definida en (4.14) para situar la base sobre el plano $z = 0$. Posteriormente, la geometría se centra en el plano XY siguiendo (4.15), garantizando una posición estable y coherente para todas las nubes generadas o procesadas.
3. **Muestreo uniforme de la superficie:** En el caso de utilizar un STL, se generan N puntos distribuidos uniformemente sobre la superficie tal y como se establece en (4.11).
4. **Voxelización de la nube de puntos:** Para reducir la densidad de puntos, se aplica una discretización mediante rejillas cúbicas siguiendo (4.16) y (4.18). El resultado es una nube voxelizada que preserva la geometría global de la pieza, reduciendo a su vez el coste computacional en las etapas posteriores.

En conjunto, este flujo transforma tanto modelos geométricos como nubes de puntos en una representación normalizada, lista para su uso en el resto del proceso. En las Figuras 4.6 y 4.7 se muestran, respectivamente, una nube generada a partir de un fichero `.stl` y una nube de puntos adquirida experimentalmente. Las Figuras 4.8a, 4.8c y 4.8b muestran la proyección de la nube de la Figura 4.7 sobre los planos XZ , YZ e XY respectivamente. Finalmente, la Tabla 4.2 resume la función de cada clase en las etapas descritas

y complementa el diagrama UML de la Figura A.2, permitiendo identificar con claridad las responsabilidades de cada componente software.

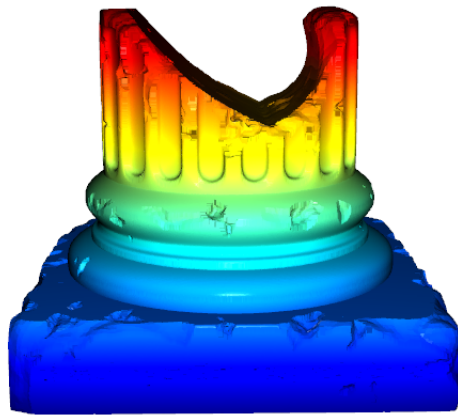
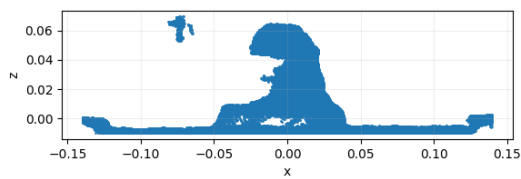


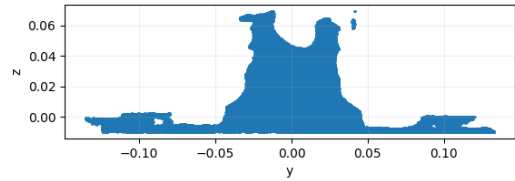
Figura 4.6 Nube de puntos generada a partir de un modelo .stl



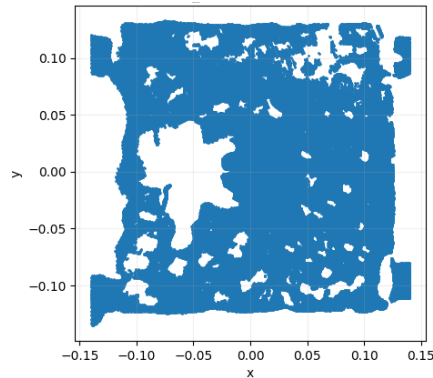
Figura 4.7 Nube de puntos adquirida experimentalmente



(a) Proyección sobre el plano XZ .



(b) Proyección sobre el plano YZ .



(c) Proyección sobre el plano XY .

Figura 4.8 Proyecciones ortogonales de la nube tras la normalización geométrica.

Tabla 4.2 Clases del módulo de representación geométrica: etapa, responsabilidades y ecuaciones.

Clase	Etapa	Responsabilidades principales	Ecuaciones
HybridPointCloud	Construcción y normalización del modelo teórico	Cargar malla STL o nube existente; convertir unidades ($\text{mm} \rightarrow \text{m}$); normalizar posición (base en $z = 0$, centrado en XY); muestrear puntos sobre la malla; generar nube voxelizada.	(4.12), (4.13), (4.14), (4.15), (4.11), (4.16), (4.18).
PointCloudProjector	Proyecciones 2D y rasterización	Proyectar puntos 3D sobre planos XY , XZ , YZ ; calcular límites y márgenes; generar histogramas bidimensionales y mapas de densidad; visualizar proyecciones.	(4.19).

4.3. Preprocesamiento de nubes

El módulo de preprocesamiento de nubes tiene como objetivo filtrar las nubes de puntos obtenidas en las etapas anteriores, de modo que resulten adecuadas para el registro geométrico y la comparación métrica entre el modelo teórico y la impresión real. A partir de nubes ya normalizadas en un mismo sistema de referencia, este módulo aplica una serie de transformaciones que reducen el ruido, eliminan puntos que no aportan información y ajustan la resolución espacial a una escala coherente con la tolerancia de registro.

En términos funcionales, el preprocesamiento combina operaciones de reducción de resolución mediante rejillas de vóxeles, filtrado estadístico y por densidad para eliminar outliers, estimación y suavizado de la geometría a través de normales, y recortes geométricos que ajustan la altura y el volumen útil de las nubes. Además, incorpora utilidades para igualar la extensión en altura del modelo teórico y de la nube generada, así como para descartar puntos ocultos en función de las posiciones de cámara registradas durante la adquisición. El resultado es un par de nubes de puntos más compactas, limpias y estructuralmente consistentes, reduciendo el riesgo de convergencias erróneas y mejorando la robustez de las métricas de validación.

4.3.1. Fundamentos teóricos

El preprocesamiento de nubes de puntos se apoya en una combinación de técnicas de reducción de resolución, detección de puntos atípicos y suavizado geométrico, complementadas con operaciones de recorte que adaptan el volumen analizado a la zona de interés. Todas estas operaciones se formulan sobre una nube de puntos tridimensional

$$\mathcal{P} = \{\mathbf{x}_k \in \mathbb{R}^3\}_{k=1}^N$$

Que ha sido normalizada previamente en un sistema de referencia común, tal y como se describió en el módulo de representación geométrica.

En primer lugar, la reducción de resolución se realiza mediante una discretización espacial basada en vóxeles. El espacio se particiona en celdas cúbicas de lado Δ y cada celda se asocia a un subconjunto de puntos \mathcal{V}_j según (4.16). La nube puede representarse entonces por un único representante por celda, típicamente el centroide

$$\mathbf{q}_j = \frac{1}{|\mathcal{V}_j|} \sum_{\mathbf{x}_k \in \mathcal{V}_j} \mathbf{x}_k$$

Generando una nube reducida $\mathcal{Q} = \{\mathbf{q}_j\}_{j=1}^{N_v}$ como en (4.18). Esta operación actúa como un filtro, eliminando detalles por debajo de la escala Δ y disminuyendo el número de

muestras sin alterar de forma significativa la estructura global de la geometría.

Para reducir la presencia de puntos atípicos debido a ruido de medida o reconstrucciones inestables, se aplican criterios de rechazo de outliers basados en la estadística local y en la densidad de vecinos. En el filtrado estadístico, para cada punto \mathbf{x}_i se considera un conjunto de vecinos \mathcal{N}_i definido mediante los k vecinos más próximos. A partir de estos vecinos se calcula la distancia media local. [37]

$$d_i = \frac{1}{|\mathcal{N}_i|} \sum_{\mathbf{x}_j \in \mathcal{N}_i} \|\mathbf{x}_i - \mathbf{x}_j\| \quad (4.20)$$

Se estima la media global μ_d y la desviación típica σ_d del conjunto $\{d_i\}_{i=1}^N$. El criterio de inlier se formula entonces como

$$d_i \leq \mu_d + \alpha \sigma_d, \quad (4.21)$$

donde α es un parámetro de tolerancia, relacionado con `std_ratio` en la implementación. Los puntos que no satisfacen (4.21) se consideran outliers y se eliminan de la nube.

Por otro lado, también se utiliza el filtrado por densidad que evalúa la cantidad de vecinos que se encuentran en un entorno esférico de radio r alrededor de cada punto. [17]

$$\mathcal{B}_i(r) = \{\mathbf{x}_j \in \mathcal{P} \mid \|\mathbf{x}_i - \mathbf{x}_j\| \leq r\} \quad (4.22)$$

Para el conjunto de vecinos dentro del radio r , el criterio de aceptación se expresa como

$$|\mathcal{B}_i(r)| \geq m \quad (4.23)$$

Donde m es el número mínimo de vecinos requerido. Los puntos que no alcanzan esta densidad mínima se eliminan, reduciendo así regiones muy dispersas que suelen corresponder a ruido.

Para poder aplicar filtros dependientes de la orientación de la superficie, es necesario disponer de una estimación de las normales en cada punto. Una manera habitual de obtenerlas consiste en aproximar localmente la superficie mediante un plano ajustado por mínimos cuadrados. Dado el vecindario \mathcal{N}_i de \mathbf{x}_i , se calcula el centroide local. [38]

$$\bar{\mathbf{x}}_i = \frac{1}{|\mathcal{N}_i|} \sum_{\mathbf{x}_j \in \mathcal{N}_i} \mathbf{x}_j$$

Y la matriz de covarianza

$$\mathbf{C}_i = \frac{1}{|\mathcal{N}_i|} \sum_{\mathbf{x}_j \in \mathcal{N}_i} (\mathbf{x}_j - \bar{\mathbf{x}}_i)(\mathbf{x}_j - \bar{\mathbf{x}}_i)^\top. \quad (4.24)$$

El vector propio, asociado al menor valor propio de \mathbf{C}_i , define la normal unitaria \mathbf{n}_i al plano que mejor se ajusta al vecindario en sentido de mínimos cuadrados.

Sobre esta estructura de normales se construye un suavizado bilateral adaptado a nubes de puntos. El objetivo es reducir el ruido preservando las discontinuidades geométricas. Para cada punto \mathbf{x}_i con normal \mathbf{n}_i y vecindario \mathcal{N}_i , se definen dos pesos para cada vecino $\mathbf{x}_j \in \mathcal{N}_i$. [39], [40]

$$w_s(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma_s^2}\right) \quad (4.25)$$

$$w_r(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{(\langle \mathbf{x}_j, \mathbf{n}_i \rangle - \langle \mathbf{x}_i, \mathbf{n}_i \rangle)^2}{2\sigma_r^2}\right) \quad (4.26)$$

Donde σ_s controla la influencia espacial y σ_r controla la sensibilidad a variaciones a lo largo de la normal. El nuevo punto suavizado se obtiene como media ponderada

$$\tilde{\mathbf{x}}_i = \frac{\sum_{\mathbf{x}_j \in \mathcal{N}_i} w_s(\mathbf{x}_i, \mathbf{x}_j) w_r(\mathbf{x}_i, \mathbf{x}_j) \mathbf{x}_j}{\sum_{\mathbf{x}_j \in \mathcal{N}_i} w_s(\mathbf{x}_i, \mathbf{x}_j) w_r(\mathbf{x}_i, \mathbf{x}_j)} \quad (4.27)$$

Tal que los puntos cercanos y con geometría compatible tienen mayor peso. Al aplicar iterativamente (4.27) sobre la nube, se atenúa el ruido de alta frecuencia sin difuminar excesivamente los bordes.

Finalmente, el preprocesamiento incorpora una serie de operaciones geométricas de recorte y ajuste de altura que actúan sobre la extensión de la nube. La medida de la cota mínima y máxima en el eje Z permite definir la altura total

$$h = z_{\text{máx}} - z_{\text{mín}} \quad (4.28)$$

Y, a partir de ella, recortar la nube a una altura objetivo h_{tar} manteniendo como base bien la cota mínima o la máxima, según convenga. De forma análoga, el recorte mediante cajas alineadas a los ejes se apoya en los límites de la caja delimitadora (4.13), restringiendo el análisis a un volumen acotado que coincide con la región relevante de la impresión.

Esto se realiza debido a que la pieza se imprime por capas, lo que implica que en ciertos puntos de impresión la geometría de la pieza y del modelo no coincidan debido a zonas internas que funcionan como soporte de la estructura. De esta forma, con el recorte se busca que la pieza que se está imprimiendo y el modelo `.stl` tengan la misma geometría al realizar la comparación.

Finalmente, también se aplica una depuración basada en visibilidad, donde sólo se conservan los puntos que resultan visibles desde una o varias posiciones de cámara regis-

tradas durante la adquisición. Este proceso puede interpretarse como una aproximación al cálculo de la envolvente visible de la nube, descartando puntos que quedarían permanentemente ocluidos y que, por tanto, no contribuyen de manera fiable a la comparación entre el modelo y la impresión. En conjunto, estas operaciones son realizadas a la nube del modelo de impresión y la nube generada, permitiendo transformar nubes ruidosas y heterogéneas en representaciones más compactas, regulares y robustas frente al ruido, adecuadas para los algoritmos de registro empleados en etapas posteriores. [41]

4.3.2. Arquitectura software del módulo

El módulo de preprocesamiento de nubes cuenta con dos componentes principales: la clase `Preprocessor`, que agrupa las operaciones de limpieza, suavizado y reducción de resolución, y la clase `CloudUtils`, que ofrece utilidades geométricas de entrada/salida para ajustar el volumen efectivo de las nubes y gestionar su visibilidad. Ambas clases se integran en la clase `Pipeline`, encargada de gestionar el flujo completo de registro entre la nube generada y el modelo teórico. La Figura A.3 recoge el diagrama UML del módulo y resume las responsabilidades de cada clase y sus relaciones.

Flujo de ejecución

El flujo de ejecución del preprocesamiento se organiza en una secuencia de operaciones que actúan sobre las nubes generada y del modelo antes de su registro. La Figura A.4 muestra el papel que desempeña este módulo dentro del *Pipeline* completo, situándose inmediatamente después de la construcción y normalización de las nubes, y justo antes de la fase de registro y comparación.

1. **Carga y recorte inicial de las nubes:** En cada iteración, la nube generada se recibe desde memoria y se normaliza mediante un recorte inicial basado en un AABB centrado, utilizando `CloudUtils.crop_aabb`. Por su parte, el modelo teórico se carga desde disco mediante `CloudUtils.load_point_cloud`, se orienta correctamente y, en caso de disponer de un fichero JSON con poses de cámara, se depura por visibilidad empleando `prune_hidden_points`. Esta función conserva únicamente los puntos visibles desde alguna o desde todas las poses disponibles, aproximando la envolvente visible del modelo.
2. **Ajuste de alturas y recorte superior:** Una vez recortadas, ambas nubes se ajustan para compartir una altura efectiva común. Para ello, primero se mide la extensión vertical de la nube generada aplicando (4.28). Este valor se usa como referencia para recortar la nube del modelo mediante `match_model_height_to_gen`. Finalmente, se elimina una delgada capa superior en ambas nubes usando `crop_from_top_mm`,

descartando zonas susceptibles de datos incompletos asociados a capas finales de la impresión.

3. **Voxelización:** El proceso de depuración continua con `Preprocessor.apply_voxel`, que aplica una discretización espacial basada en vóxeles siguiendo el esquema definido en (4.16)–(4.18). Esta operación reduce de forma controlada el número de puntos.
4. **Filtrado estadístico:** A continuación, `apply_statistical` elimina puntos atípicos basándose en la distancia media local a los vecinos más próximos, formulada en (4.20). El criterio de aceptación se establece según el umbral (4.21), permitiendo conservar únicamente aquellos puntos cuya coherencia local se ajusta al comportamiento esperado de la superficie.
5. **Estimación de normales y suavizado bilateral:** Para poder aplicar un suavizado coherente con la geometría local, se emplea `ensure_normals`, que estima las normales en cada punto mediante un ajuste local de plano, utilizando la matriz de covarianza definida en (4.24). Sobre esta estructura se aplica `bilateral_filter`, que implementa un suavizado bilateral sobre la superficie: los pesos espaciales se calculan conforme a (4.25) y los pesos de rango siguiendo (4.26). La actualización de cada punto se realiza con la media ponderada dada en (4.27), reduciendo el ruido de alta frecuencia sin borrar los detalles geométricos relevantes.
6. **Filtrado por densidad:** Tras el suavizado, `apply_density` elimina regiones excesivamente dispersas aplicando el criterio de densidad mínima expresado en (4.23). Esta operación refuerza la coherencia estructural de la nube al eliminar puntos aislados producto de ruido de profundidad o reconstrucciones incompletas.
7. **Flujo de preprocesamiento completo mediante `enhanced`:** Aunque cada operación puede emplearse de forma independiente, el método `Preprocessor.enhanced` encapsula un flujo ajustado específicamente para las nubes generada y teórica. Para la nube generada se aplica una voxelización inicial, un filtrado estadístico más estricto, un suavizado bilateral con parámetros más agresivos y un filtrado por densidad. En el caso del modelo, los parámetros son más moderados, consistentes con su mayor regularidad geométrica. Este flujo produce un par de nubes limpias, uniformes y estables que se entregan al módulo de registro.

En conjunto, este flujo de ejecución transforma nubes ruidosas en representaciones geométricas consistentes y depuradas. El resultado final constituye la entrada al módulo de registro, donde la calidad del preprocesamiento es relevante para alcanzar una convergencia y unos valores métricos fiables en el análisis de la impresión. La Tabla 4.3 resume

las operaciones principales aplicadas en esta etapa, junto con las responsabilidades de cada componente y los criterios teóricos asociados. Por otro lado, las Figuras 4.9 y ?? ilustran un esquema del flujo seguido para la nube generada experimentalmente y para la nube obtenida a partir del modelo .stl respectivamente.

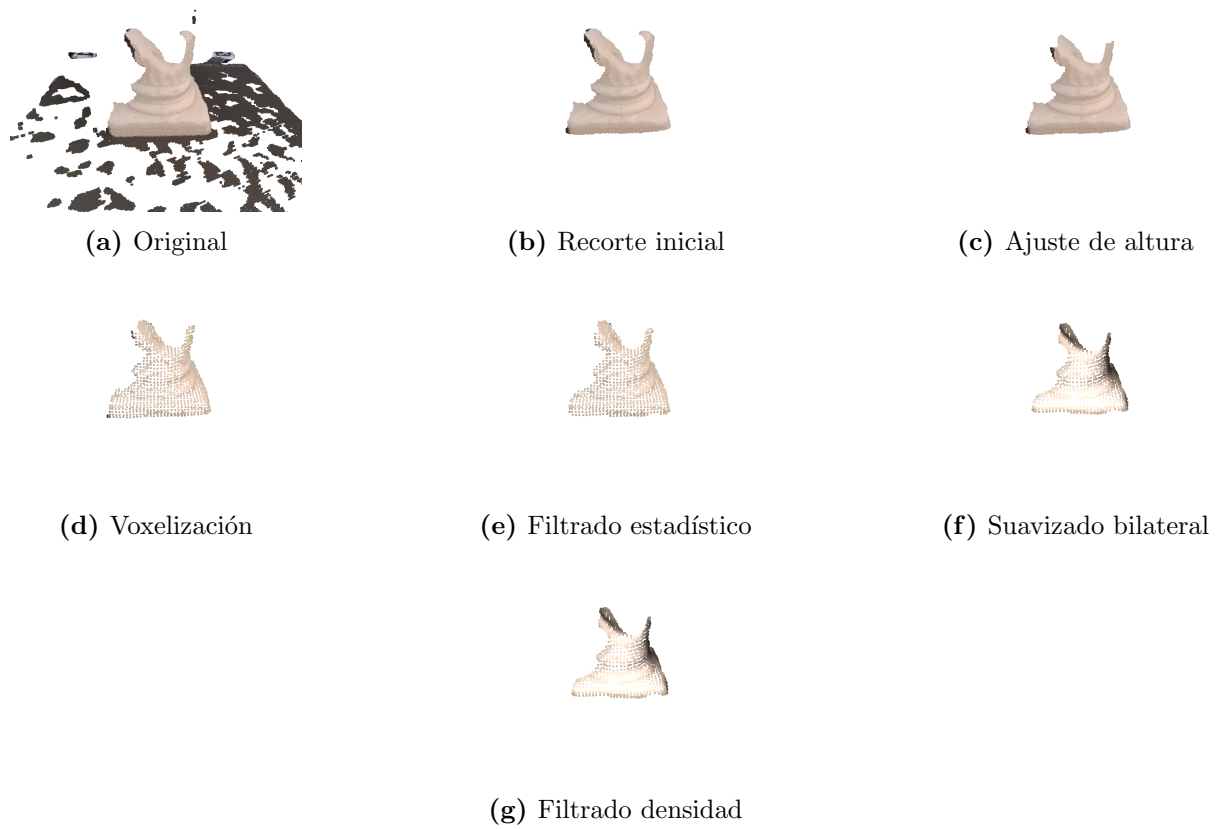


Figura 4.9 Flujo de preprocesamiento para nube de puntos generada experimentalmente

Tabla 4.3 Resumen de operaciones del preprocesamiento: etapas, responsabilidades y criterios aplicados.

Componente	Etapas	Responsabilidades principales	Ecuaciones / criterios
Apply_voxel	Reducción de resolución	Agrupar puntos en vóxeles de lado Δ y los reemplaza por su centroide.	(4.16)–(4.18).
Apply_statistical	Filtrado estadístico	Elimina puntos con distancia media local anómala respecto a vecinos cercanos.	(4.20), (4.21).
Apply_density	Filtrado por densidad	Descarta puntos con menos de m vecinos dentro de un radio r .	(4.23).
Ensure_normals	Estimación de normales	Calcula normales mediante el vector propio asociado al menor autovalor de la covarianza local.	(4.24).
Bilateral_filter	Suavizado bilateral	Suaviza manteniendo bordes: pesos espaciales y de rango mediante gaussianas; actualización ponderada.	(4.25)–(4.27).
Enhanced	Flujo completo	Encadena voxel, estadístico, bilateral, densidad con parámetros diferenciados para nube generada y modelo.	Todas las anteriores.
Prune_hidden_points	Depuración por visibilidad	Conserva puntos visibles desde una o varias cámaras según poses en JSON.	Criterios de visibilidad.
Crop_aabb	Recorte volumétrico	Recorta por un AABB alineado a ejes, limitando el espacio útil de análisis.	(4.13).
Measure_height_z	Cálculo de altura	Obtiene z_{\min} , z_{\max} y altura total h .	(4.28).
Match_model_height_to_gen	Ajuste de alturas	Recorta el modelo para igualar su altura a la nube generada.	(4.28).
Crop_from_top_mm	Recorte superior	Elimina un espesor en mm desde la parte superior para descartar zonas no fiables.	(4.28).

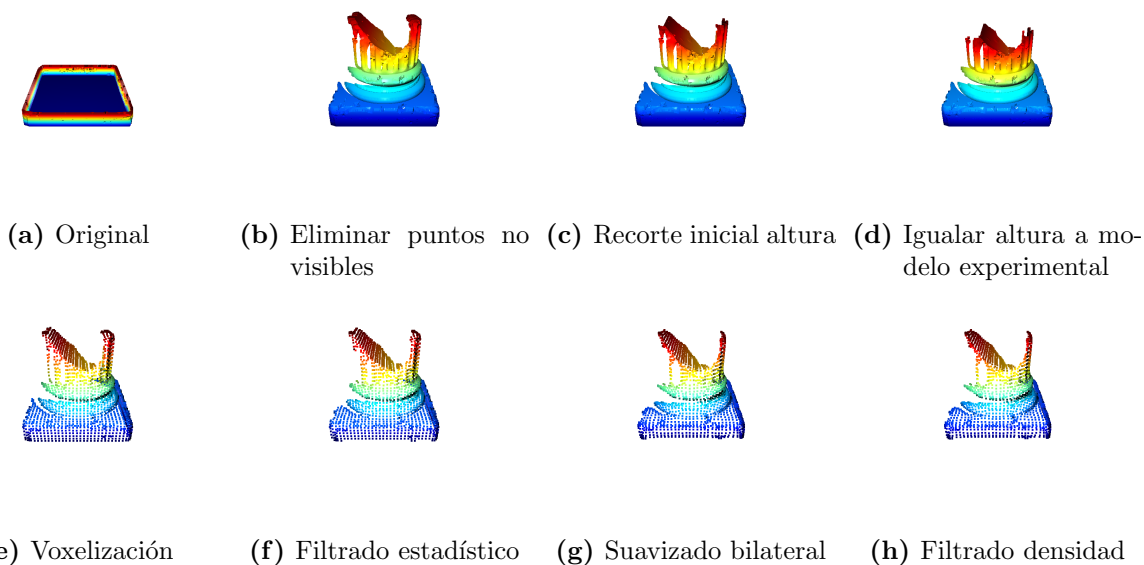


Figura 4.10 Flujo de preprocesamiento para nube de puntos generada de un archivo `.stl`

4.4. Registro de nubes de puntos

El módulo de registro de nubes de puntos tiene como objetivo alinear, en un mismo marco de referencia, la nube generada durante la adquisición y el modelo teórico de la pieza. A partir de las nubes ya normalizadas y preprocesadas en los módulos anteriores, esta etapa estima una transformación rígida formada por una matriz de rotación $\mathbf{R} \in SO(3)$ y un vector de traslación $\mathbf{t} \in \mathbb{R}^3$ que aproxima la geometría observada a la geometría ideal.

Desde el punto de vista funcional, el módulo combina distintos niveles de registro. En primer lugar, se construye un par de nubes a partir de la salida del preprocesamiento, ajustando recortes, alturas y visibilidad de acuerdo con la zona de interés. A continuación, se ejecuta un método de registro llamado CliReg, que se encarga de obtener una solución inicial fiable incluso en presencia de ruido, oclusiones parciales y solapes limitados entre las nubes. Sobre esta base, el sistema puede incorporar refinamientos locales mediante algoritmos tipo ICP y comparar el rendimiento de distintos métodos clásicos de registro global, RANSAC y FGR, sobre las mismas nubes.

El resultado de esta etapa es una pareja de nubes alineadas y un conjunto de métricas de convergencia que cuantifican la calidad del registro: error medio cuadrático, solapamiento geométrico y porcentaje de inliers. Estas salidas constituyen la entrada directa al módulo de evaluación de impresión, donde se analizan tanto el porcentaje de superficie cubierta como el avance en altura de la pieza a lo largo del proceso de fabricación.

4.4.1. Fundamentos teóricos del registro

El problema de registro rígido entre dos nubes de puntos se formula a partir de dos conjuntos tridimensionales

$$\mathcal{G} = \{\mathbf{g}_i \in \mathbb{R}^3\}_{i=1}^{N_G}, \quad \mathcal{M} = \{\mathbf{m}_i \in \mathbb{R}^3\}_{i=1}^{N_M}$$

que representan respectivamente la nube generada y la nube del modelo, ambas expresadas en el mismo sistema de referencia. El objetivo es encontrar una transformación rígida (\mathbf{R}, \mathbf{t}) tal que la imagen transformada de \mathcal{G} se aproxime a \mathcal{M} .

Planteamiento clásico del registro rígido

Asumiendo que se dispone de un conjunto de correspondencias \mathcal{C} entre puntos de ambas nubes,

$$\mathcal{C} = \{(i, j) \mid \mathbf{g}_i \leftrightarrow \mathbf{m}_j\},$$

el problema puede escribirse como la minimización de una energía de tipo punto-a-punto:

$$\min_{\mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3} E(\mathbf{R}, \mathbf{t}) = \sum_{(i,j) \in \mathcal{C}} \|\mathbf{R} \mathbf{g}_i + \mathbf{t} - \mathbf{m}_j\|^2. \quad (4.29)$$

La restricción $\mathbf{R} \in SO(3)$ impone que la transformación preserve distancias y ángulos, de modo que únicamente se permiten rotaciones y traslaciones, sin cambios de escala ni deformaciones. Para un conjunto de correspondencias fijo, el problema (4.29) admite una solución cerrada basada en la descomposición en valores singulares de la matriz de correlación entre los puntos emparejados. [27]

En la práctica, sin embargo, las correspondencias son desconocidas y deben ser estimadas a partir de las propias nubes. Esta dificultad es especialmente relevante en el contexto de impresión 3D, donde la nube generada puede contener únicamente una parte de la pieza, estar contaminada por ruido de profundidad y presentar zonas ocluidas.

Registro global basado en correspondencias dispersas

Un enfoque habitual para estimar correspondencias iniciales consiste en trabajar con un subconjunto de puntos salientes o *keypoints* y describir su entorno mediante descriptores locales invariantes a rotaciones. En este trabajo, estos conceptos se materializan a través de la extracción de puntos característicos y la computación de descriptores FPFH, *Fast Point Feature Histograms*, que codifican la distribución angular de las normales en un vecindario de cada punto. [17]

A partir de estos descriptores, se buscan correspondencias potenciales entre \mathcal{G} y \mathcal{M} por similitud en el espacio de características. El conjunto resultante de pares suele contener una proporción significativa de correspondencias erróneas (*outliers*), por lo que es necesario aplicar esquemas de estimación a partir de subconjuntos.

Métodos como RANSAC, *Random Sample Consensus*, y FGR, *Fast Global Registration*, abordan este problema seleccionando subconjuntos reducidos de correspondencias, probando hipótesis de transformación y evaluando su coherencia global. En términos generales, estos métodos buscan una transformación (\mathbf{R}, \mathbf{t}) que maximice el número de correspondencias que quedan dentro de una tolerancia geométrica τ , o que minimice una energía robusta sobre las distancias residuales. [28]

$$E_{\text{rob}}(\mathbf{R}, \mathbf{t}) = \sum_{(i,j) \in \mathcal{C}} \rho(\|\mathbf{R} \mathbf{g}_i + \mathbf{t} - \mathbf{m}_j\|; \tau) \quad (4.30)$$

Donde $\rho(\cdot; \tau)$ es una función de coste robusta que atenúa la influencia de los outliers. El resultado de esta etapa es una transformación inicial razonable incluso cuando la proporción de correspondencias erróneas es elevada.

Refinamiento local mediante ICP

Una vez disponible una transformación inicial, es posible refinarla utilizando algoritmos locales de registro denso, siendo el método ICP, *Iterative Closest Point*, uno de los más extendidos. ICP alterna dos pasos principales: construcción de correspondencias por vecindad más cercana entre los puntos de la nube transformada y la nube de referencia, y resolución del problema de mínimos cuadrados (4.29) dadas esas correspondencias. [28]

Una variante especialmente adecuada para nubes con superficie bien definida es ICP punto-a-plano, en la que el error se mide proyectando la diferencia entre puntos sobre la normal de la superficie en el modelo. Dado un conjunto de correspondencias (i, j) y las normales \mathbf{n}_j en la nube de referencia, el problema se formula como:

$$\min_{\mathbf{R}, \mathbf{t}} \sum_{(i,j) \in \mathcal{C}} (\mathbf{n}_j^\top (\mathbf{R} \mathbf{g}_i + \mathbf{t} - \mathbf{m}_j))^2 \quad (4.31)$$

Lo que conduce a una actualización iterativa de (\mathbf{R}, \mathbf{t}) hasta convergencia. Este refinamiento mejora la alineación local siempre que la inicialización se encuentre suficientemente cerca de la solución óptima.

Registro con algoritmo CliReg

El método CliReg aborda el problema de las correspondencias desde una perspectiva de grafos. En lugar de evaluar cada correspondencia de forma aislada, se construye un grafo donde los nodos representan correspondencias candidato y las aristas conectan pares de correspondencias cuyas distancias y ángulos son compatibles entre ambas nubes. Sobre este grafo se buscan *cliques* máximos, esto es, subconjuntos de correspondencias completamente conectadas entre sí. [30]

De manera simplificada, CliReg persigue seleccionar un subconjunto $\mathcal{C}^* \subset \mathcal{C}$ de correspondencias que maximiza simultáneamente su cardinalidad y su coherencia geométrica. [30]

$$\mathcal{C}^* = \arg \max_{\mathcal{S} \subset \mathcal{C}} |\mathcal{S}| \quad \text{sujeto a que } \mathcal{S} \text{ forme un cliqué consistente.} \quad (4.32)$$

A partir de este subconjunto consistente, se estima la transformación rígida resolviendo (4.29) restringida a \mathcal{C}^* . Gracias a esta formulación, el método resulta especialmente robusto frente a altos porcentajes de outliers, escenarios de solapamiento parcial y variaciones locales de densidad, condiciones habituales en las nubes generadas durante el proceso de impresión 3D. [30]

En conjunto, la combinación de registro global (RANSAC/FGR), refinamiento local mediante ICP y selección robusta de correspondencias mediante CliReg proporciona un marco sólido para alinear la nube generada con el modelo teórico. Sobre esta base se construyen las métricas de convergencia y las medidas de progreso que se utilizarán en el módulo de evaluación de la impresión. [30]

4.4.2. Arquitectura software del módulo

El módulo de registro se estructura con un conjunto de clases que encapsulan las distintas funciones de alineamiento entre nubes de puntos. Su diseño sigue una arquitectura modular, donde cada componente ejecuta una parte concreta del proceso: estimación de correspondencias, registro inicial, refinamiento local e integración final de resultados. La Figura A.5 se muestra el diagrama UML correspondiente, donde se aprecia la relación entre los distintos métodos de registro, la clase base que los unifica y el bloque específico dedicado a CliReg.

La arquitectura se divide en cuatro niveles funcionales: extracción de características y preparación de datos, métodos clásicos de registro global y refinamiento local, registro mediante CliReg y coordinación del proceso dentro de A.4. A continuación se describen cada uno de estos niveles y las clases que los implementan.

Extracción de características y construcción de correspondencias

El primer bloque está compuesto por la clase `FeatureUtils`, responsable de generar la información geométrica necesaria para los métodos de registro global. Esta clase implementa dos operaciones principales:

- **Cálculo de normales:** se estiman las normales locales para cada punto, garantizando una representación geométrica uniforme y apta para algoritmos basados en descriptores.
- **Cálculo de descriptores FPFH:** se generan histogramas de características que codifican la estructura del vecindario 3D, permitiendo establecer correspondencias iniciales entre la nube generada y el modelo.

Esta fase actúa como preparación para los métodos RANSAC, FGR y CliReg, que requieren correspondencias o características locales para sus estimaciones iniciales.

Jerarquía de métodos de registro

El segundo nivel de la arquitectura lo ocupa la clase abstracta `RegistrationMethod`, que define una interfaz común para todos los algoritmos de registro implementados. Cada método concreto hereda de esta clase y sobrescribe la operación principal `run()`, encargada de devolver una estimación de la matriz de transformación rígida acompañada del tiempo total de ejecución.

Los métodos derivados incluyen:

- **RANSACMethod:** implementa un esquema de registro global basado en la verificación robusta de correspondencias dispersas. Evalúa hipótesis de transformación formadas a partir de subconjuntos aleatorios y selecciona aquella con mayor número de inliers.
- **RANSACPlusICP:** combina una inicialización RANSAC con una etapa posterior de refinamiento mediante ICP punto-a-plano, integrando precisión local.
- **FGRMethod:** implementa *Fast Global Registration*, un algoritmo que optimiza energía sobre correspondencias, proporcionando convergencia rápida sin depender de muestreos aleatorios.
- **FGRPlusICP:** similar al caso anterior, añade una fase de ICP tras la inicialización mediante FGR para mejorar la alineación fina.

- **ICPRefineMethod:** encapsula un refinamiento local tipo ICP, aplicable como etapa independiente o como complemento de otros métodos.

Esta jerarquía permite evaluar de forma uniforme distintos enfoques de registro y comparar su rendimiento bajo las mismas condiciones.

Registro mediante `CliRegMatcher`

El núcleo del módulo es la clase `CliRegMatcher`, que actúa como envoltorio del algoritmo `CliReg` de correspondencias por cliques. Esta clase recibe como entrada dos nubes de puntos densas y realiza las siguientes operaciones:

1. **Preprocesamiento interno:** cálculo de normales, extracción opcional de keypoints y cómputo de descriptores locales.
2. **Construcción del grafo de correspondencias:** cada correspondencia se modela como un nodo, y las aristas reflejan compatibilidad geométrica basada en distancias y orientaciones relativas.
3. **Búsqueda del cliqué consistente de mayor tamaño:** se selecciona el subconjunto de correspondencias más coherente estructuralmente, mitigando la presencia de outliers y casos de solapamiento parcial.
4. **Estimación de la transformación rígida:** se obtiene la matriz \mathbf{R} y el vector \mathbf{t} .

La clase provee además de métricas internas tales como la tasa de correspondencias válidas, número de vecinos consistentes y detalles del proceso de grafo, que resultan útiles para análisis y depuración.

Flujo de ejecución

El flujo de ejecución del módulo de registro sigue una secuencia que abarca la estimación de una transformación inicial robusta y el refinamiento final de la alineación. Este proceso es llevado a cabo por la clase `Pipeline` de la Figura A.4, que integra las salidas del preprocesamiento con las distintas estrategias de registro disponibles. A continuación se describe cada una de las etapas implicadas:

1. **Recepción de nubes preprocesadas.** La ejecución comienza una vez que la nube generada y el modelo teórico han sido normalizados, filtrados y recortados por el módulo de preprocesamiento. En este punto, ambas nubes comparten el mismo sistema de referencia y tienen una densidad espacial controlada. `.single_iteration` recibe estas nubes listas para ser alineadas.

2. **Registro por CliReg.** Se ejecuta el algoritmo principal de registro a través de `CliReg_icp`.

- `CliRegMatcher` calcula descriptores, genera correspondencias candidato y construye un grafo de compatibilidad geométrica.
- A partir de este subconjunto se estima una transformación rígida que sirve como solución inicial o como alineación final, dependiendo de si se activa o no una etapa posterior de ICP.

En la Figura 4.11 se observa un esquemático de cómo es el flujo para el registro con CliReg. Donde la nube azul es la obtenida experimentalmente, la nube roja del modelo de impresión y la zonas violetas la superposición entre ambas nubes de puntos.

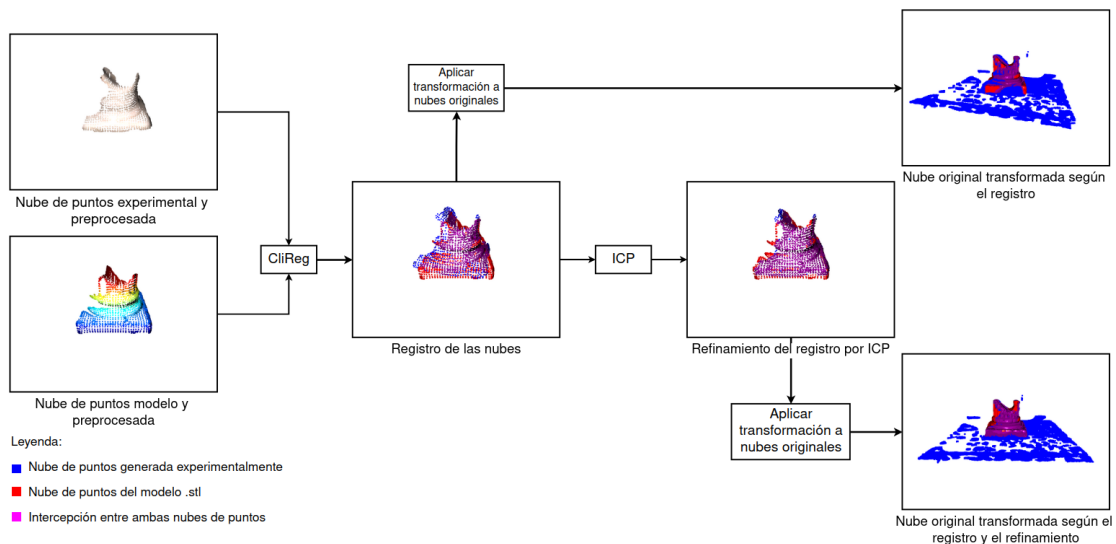


Figura 4.11 Registro de las nubes de puntos con el algoritmo CliReg

3. **Evaluación de métodos alternativos.** La clase `Comparator` ejecuta en paralelo distintos métodos de registro derivados de `RegistrationMethod`, tales como `RANSACMethod`, `FGRMethod`, `RANSACPlusICP` y `FGRPlusICP`. Cada método devuelve su propia transformación y métricas de convergencia, permitiendo evaluar la precisión de cada aproximación. En las Figuras y se observa el proceso para RANSAC y FGR respectivamente.

4. **Selección del mejor método y consolidación de resultados.** El módulo selecciona la transformación que ofrezca el equilibrio entre precisión y estabilidad. Esta transformación final se aplica a la nube generada, produciendo el par de nubes alineado que será analizado posteriormente en el módulo de evaluación de impresión.

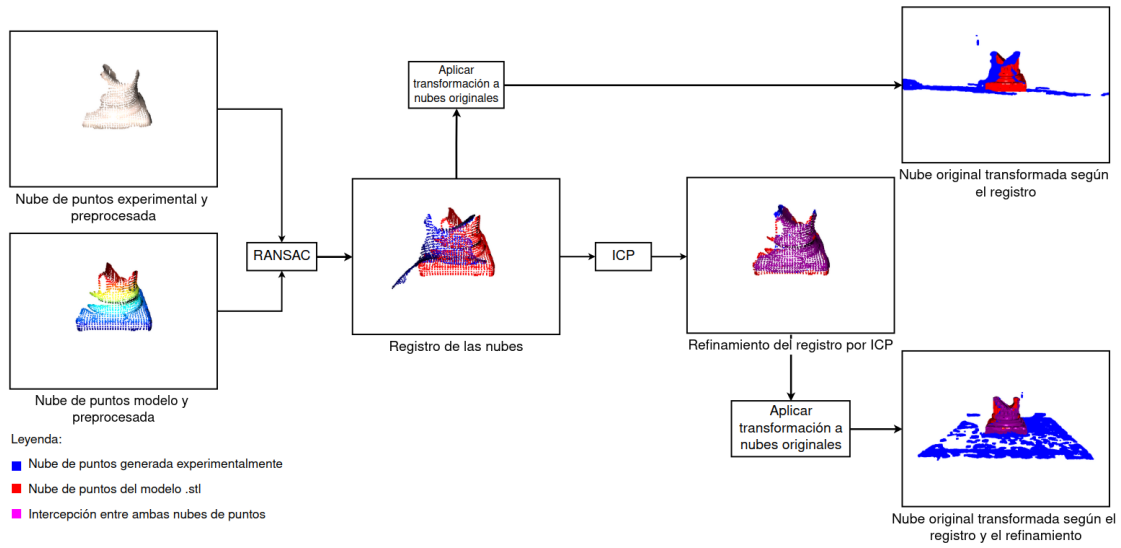


Figura 4.12 Registro de las nubes de puntos con el algoritmo RANSAC

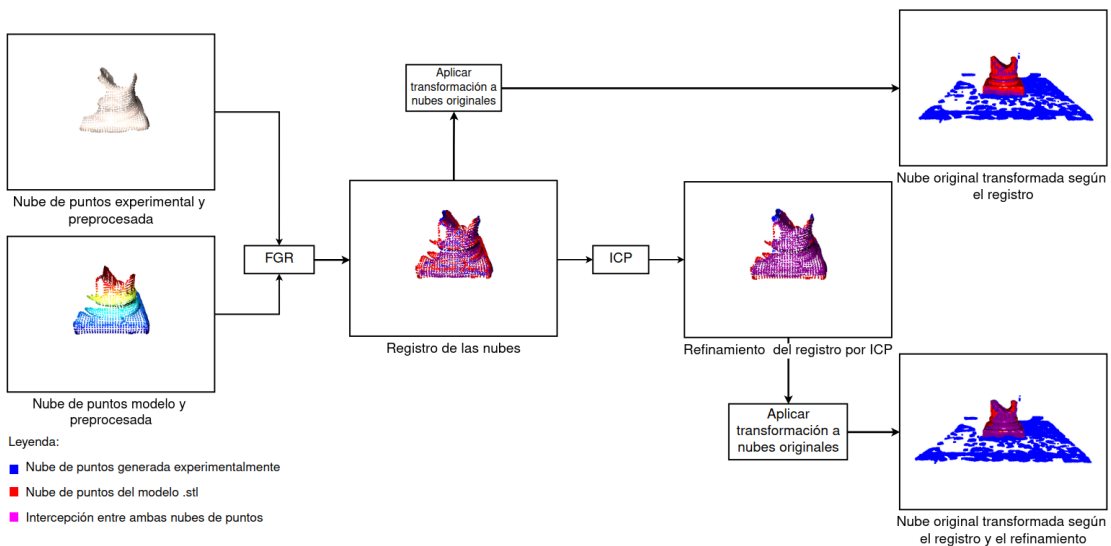


Figura 4.13 Registro de las nubes de puntos con el algoritmo FGR

Este flujo garantiza que el proceso de registro sea modular y reproducible, permitiendo comparar distintas técnicas bajo un mismo marco y asegurar la coherencia necesaria para las métricas de validación.

La Tabla 4.4 resume las operaciones ejecutadas durante el proceso de registro, organizando cada componente según su responsabilidad dentro del flujo descrito. En ella se identifican las etapas clave junto con las responsabilidades específicas de cada clase y los criterios teóricos que sustentan su funcionamiento. Esta visión resumida facilita comprender cómo se integran los distintos métodos dentro del módulo y cómo contribuyen a obtener una alineación entre la nube generada y el modelo teórico.

Tabla 4.4 Resumen de operaciones del módulo de registro: etapas, responsabilidades y criterios geométricos.

Componente	Etapas	Responsabilidades principales	Criterios / ecuaciones
FeatureUtils	Preparación de características	Calcular normales y extraer descriptores FPFH para la generación de correspondencias dispersas.	Distancias locales; histogramas FPFH; vecindario k -NN.
RANSACMethod	Registro global robusto	Probar subconjuntos aleatorios de correspondencias y estimar transformaciones iniciales robustas a outliers.	Minimización de (4.30); umbral de inliers τ .
FGRMethod	Registro global eficiente	Optimizar una energía sobre correspondencias sin muestreo aleatorio, proporcionando convergencia rápida.	Optimización robusta de (4.30).
ICPRefineMethod	Refinamiento local	Ajustar la solución inicial mediante ICP punto-a-plano para mejorar la precisión fina del alineamiento.	Error puntual-a-plano (4.31).
RANSACPlusICP	Inicialización + refinamiento	Combinar RANSAC con ICP.	(4.30) seguido de (4.31).
FGRPlusICP	Inicialización + refinamiento	Combinar FGR con refinamiento ICP.	Optimización de (4.30) seguida de (4.31).
CliRegMatcher	Registro robusto por cliques	Construir el grafo de correspondencias compatibles, detectar cliques consistentes y estimar la transformación rígida a partir del subconjunto geoméricamente coherente.	Problema de cliqué (4.32); estimación rígida (4.29).
Pipeline	Coordinación del registro	Integrar todas las etapas: selección del método, ejecución del registro, refinamiento opcional y devolución del par de nubes alineado.	Criterios de convergencia.

4.5. Evaluación y comparación

El módulo de evaluación y comparación tiene como objetivo cuantificar la calidad de la alineación obtenida durante el registro, así como estimar el grado de avance de la impresión 3D a partir de la nube generada. Este módulo actúa como etapa final del sistema y proporciona las métricas necesarias para validar si la geometría observada coincide con la geometría teórica, tanto en términos de ajuste superficial como de completitud volumétrica.

En particular, este módulo integra tres tipos de métricas complementarias. En primer lugar, las métricas de convergencia evalúan la consistencia geométrica entre dos nubes ya alineadas, midiendo distancias punto-a-punto, solapamiento, porcentaje de inliers y medidas de error. En segundo lugar, las métricas de progreso de impresión estiman cuánto de la superficie del modelo ha sido efectivamente reproducida, así como el avance en altura en relación con la geometría teórica. Finalmente, el módulo incorpora métricas de proyección en el plano YZ, que permiten evaluar el solapamiento aparente entre ambas nubes y generan representaciones visuales que ayudan a la inspección cualitativa del registro. Además, el módulo permite generar informes en formato CSV con las métricas obtenidas, facilitando su análisis externo y la documentación del comportamiento del sistema.

En conjunto, este módulo cierra el flujo funcional del pipeline y proporciona un criterio cuantitativo para validar la precisión del registro y el progreso real de la impresión.

4.5.1. Fundamentos teóricos

Las métricas empleadas se basan en distancias de vecindad más cercana, criterios de coincidencia bajo tolerancia y estimadores de progreso según la geometría del modelo teórico. A continuación se describen los fundamentos que sustentan cada conjunto de métricas.

Distancias punto a punto y métricas de convergencia

Sea $\mathcal{A} = \{\mathbf{a}_i\}_{i=1}^{N_A}$ la nube generada alineada y $\mathcal{B} = \{\mathbf{b}_j\}_{j=1}^{N_B}$ la nube del modelo. El análisis de convergencia parte del cálculo de distancias de Vecino Más Cercano (NN) en ambos sentidos. Para cada punto \mathbf{a}_i se define: [17]

$$d_{A \rightarrow B}(i) = \min_j \|\mathbf{a}_i - \mathbf{b}_j\|. \quad (4.33)$$

De igual forma, para cada punto \mathbf{b}_j del modelo se define:

$$d_{B \rightarrow A}(j) = \min_i \|\mathbf{b}_j - \mathbf{a}_i\|. \quad (4.34)$$

Estas distancias permiten construir la *Raíz del Error Cuadrático Medio (RMSE)*, que resume en una sola magnitud la coherencia global del alineamiento al penalizar desviaciones en ambos sentidos:

$$\text{RMSE}_{\text{sym}} = \sqrt{\frac{1}{N_A + N_B} \left(\sum_{i=1}^{N_A} d_{A \rightarrow B}(i)^2 + \sum_{j=1}^{N_B} d_{B \rightarrow A}(j)^2 \right)}. \quad (4.35)$$

En este contexto, una alineación de buena calidad produce valores bajos de RMSE_{sym} y distribuciones de NN concentradas, mientras que desalineaciones locales o sistemáticas se manifiestan como colas largas y aumentos de la dispersión.

Además, se define un porcentaje de solapamiento, *overlap*, como la proporción de puntos de la nube generada que se consideran *inliers* respecto al modelo bajo una tolerancia geométrica τ . [17]

$$\text{overlap} = \frac{|\{i \mid d_{A \rightarrow B}(i) \leq \tau\}|}{N_A}. \quad (4.36)$$

En la práctica, esta magnitud suele reportarse en porcentaje multiplicando por 100. Un valor alto de *overlap* indica que una gran fracción de la nube generada coincide con la superficie teórica dentro de la tolerancia, proporcionando un criterio directo de consistencia geométrica tras el registro.

Cobertura superficial del modelo: *progress surface*

Con el fin de estimar el avance real de la impresión, se aplica una métrica que evalúa qué proporción de la superficie teórica ha sido explicada por la nube generada. Para cada punto $\mathbf{m} \in \mathcal{M}$ del modelo, se calcula su distancia al punto más cercano de la nube generada \mathcal{G} . [42]

$$d(\mathbf{m}) = \min_{\mathbf{g} \in \mathcal{G}} \|\mathbf{m} - \mathbf{g}\|. \quad (4.37)$$

La cobertura superficial, denominada también *progress surface* en la implementación, se define como la fracción de puntos del modelo que encuentran soporte en la nube generada dentro del umbral τ : [42]

$$\text{coverage} = \frac{|\{\mathbf{m} \in \mathcal{M} \mid d(\mathbf{m}) \leq \tau\}|}{|\mathcal{M}|}. \quad (4.38)$$

Esta métrica refleja el porcentaje del modelo que ha sido efectivamente reconstruido, resultando especialmente informativa para estimar progreso por capas incluso cuando la nube generada no cubre uniformemente toda la geometría.

Completitud de la nube generada respecto al modelo: *perc b in a*

De forma complementaria a la cobertura del modelo, es relevante cuantificar qué fracción de la nube generada puede explicarse por el modelo, penalizando puntos que aparecen fuera del volumen teórico. Para ello se define la distancia del punto generado a su vecino más cercano en el modelo:

$$d_{\text{vol}}(\mathbf{g}) = \min_{\mathbf{m} \in \mathcal{M}} \|\mathbf{g} - \mathbf{m}\|. \quad (4.39)$$

A partir de esta distancia, la métrica *perc b in a* se define como:

$$\text{perc b in a} = \frac{|\{\mathbf{g} \in \mathcal{G} \mid d_{\text{vol}}(\mathbf{g}) \leq \tau\}|}{|\mathcal{G}|}. \quad (4.40)$$

Al igual que el solapamiento, suele reportarse en porcentaje multiplicando por 100. Mientras que (4.38) mide cuánto de la superficie modelo está presente, (4.40) mide cuánto del volumen observado es consistente con el modelo, aportando un criterio bidimensional y tridimensional.

Progreso en altura: *zmax percent*

Además de la cobertura, resulta útil estimar el avance en altura de la impresión. Para una nube cualquiera se define la altura mediante su caja delimitadora alineada a ejes:

$$h = z_{\text{máx}} - z_{\text{mín}}. \quad (4.41)$$

Una forma directa de expresar el avance global en altura es el cociente entre la altura observada y la altura total del modelo:

$$\text{progress}_z = \frac{h_{\text{gen}}}{h_{\text{model}}}. \quad (4.42)$$

No obstante, dado que en impresión por capas el indicador más estable suele ser el máximo alcanzado en z , se define también *zmax percent* como:

$$\text{zmax percent} = \frac{z_{\text{máx}}(\mathcal{G}) - z_{\text{mín}}(\mathcal{M})}{z_{\text{máx}}(\mathcal{M}) - z_{\text{mín}}(\mathcal{M})}. \quad (4.43)$$

En configuraciones habituales donde $z_{\min}(\mathcal{M}) \approx 0$, la métrica se reduce a $z_{\max}(\mathcal{G})/h_{\text{model}}$. Al igual que las anteriores, suele reportarse en porcentaje multiplicando por 100.

Proyección y solapamiento en el plano YZ

Para complementar el análisis tridimensional, se genera una proyección ortogonal sobre el plano YZ . [43]

$$\pi_{YZ}(\mathbf{x}) = \begin{bmatrix} Y \\ Z \end{bmatrix}. \quad (4.44)$$

A partir de esta proyección, se calcula un solapamiento bidimensional entre las ocupaciones proyectadas de ambas nubes. Sea Ω_A y Ω_B el conjunto de celdas (o píxeles) ocupadas por $\pi_{YZ}(\mathcal{A})$ y $\pi_{YZ}(\mathcal{B})$, respectivamente. Entonces:

$$\text{overlap}_{YZ} = \frac{|\Omega_A \cap \Omega_B|}{|\Omega_A \cup \Omega_B|}. \quad (4.45)$$

Esta métrica permite evaluar visual y cuantitativamente la coherencia del registro en un plano especialmente informativo para la impresión, donde la altura y el contorno lateral son determinantes.

En conjunto, estas formulaciones permiten evaluar el registro y el progreso de impresión mediante criterios cuantitativos y reproducibles, que se referencian posteriormente en el capítulo de Resultados.

4.5.2. Arquitectura software del módulo

El módulo de evaluación y comparación se organiza en torno a cinco componentes principales, cada uno encargado de calcular un conjunto específico de métricas o de gestionar la exportación de los resultados. La Figura A.6 muestra el diagrama UML correspondiente, donde se representan las clases involucradas y las relaciones que mantienen entre sí.

En primer lugar, la clase `Comparator` actúa como núcleo del módulo y coordina la evaluación de las transformaciones obtenidas durante el registro. Esta clase recibe como entrada el conjunto de transformaciones propuestas por los distintos métodos y aplica el cálculo de métricas de convergencia mediante el módulo `convergence_metrics`. Cada evaluación se empaqueta en una instancia de `MethodResult`, que contiene la transformación aplicada, la nube alineada, el tiempo de ejecución y un diccionario con todas las métricas cuantitativas asociadas.

Para completar el análisis geométrico, la clase `SurfaceCoverageProgress` implementa el cálculo de cobertura superficial del modelo (4.38), la completitud de la nube generada respecto al modelo (4.40), así como el progreso en altura (4.42) y el avance máximo en z (4.43). Estas métricas permiten establecer una medida objetiva sobre el estado de la impresión, independiente del método de registro.

Por otro lado, la clase `YZPlotter` genera representaciones bidimensionales en el plano YZ a partir de (4.44), calcula el solapamiento proyectado (4.45) y produce una imagen ilustrativa que facilita la inspección lateral del contorno y la altura.

Finalmente, la clase `MetricsExporter` centraliza el proceso de exportación de métricas, generando archivos `CSV` y descripciones auxiliares. Este componente permite documentar cada iteración del pipeline y facilita la comparación entre distintos métodos de registro y ejecuciones sucesivas.

En conjunto, estas clases proporcionan una arquitectura modular, extensible y completamente integrada con las etapas anteriores del sistema, permitiendo evaluar de manera cuantitativa y reproducible la calidad del registro y el progreso real de la impresión.

Flujo de ejecución

El proceso completo de evaluación y comparación sigue una secuencia que integra las métricas de registro, de progreso geométrico y de solapamiento proyectado. El flujo puede describirse a través de las siguientes etapas:

1. **Recepción de las nubes alineadas y de los resultados de registro.** La clase `Pipeline` entrega a `Comparator` las nubes generada y del modelo ya preprocesadas, junto con las transformaciones propuestas por los distintos métodos de registro evaluados.
2. **Cálculo de métricas de convergencia.** Para cada método, `Comparator` aplica `compute_convergence` del módulo `convergence_metrics`, obteniendo distancias NN (4.33)–(4.34), el error RMSE_{sym} (4.35) y el solapamiento (4.36), además de medidas estadísticas asociadas a dichas distribuciones.
3. **Evaluación del progreso de impresión.** La clase `SurfaceCoverageProgress` estima la cobertura superficial del modelo (4.38), la completitud de la nube generada respecto al modelo (4.40), así como el progreso en altura (4.42) y el avance máximo alcanzado en el eje z (4.43).
4. **Cálculo del solapamiento en el plano YZ .** `YZPlotter` proyecta ambas nubes sobre el plano YZ (4.44), calcula el solapamiento proyectado (4.45) y genera una

imagen ilustrativa que refleja las diferencias visibles entre el contorno ideal y el observado.

5. **Consolidación de resultados y construcción de `MethodResult`.** Cada conjunto de métricas se encapsula en una instancia de `MethodResult`, que incluye el nombre del método, la transformación aplicada, la nube alineada, el tiempo de ejecución y el diccionario completo de métricas calculadas.
6. **Exportación de resultados.** Finalmente, `MetricsExporter` recibe la lista de resultados y genera ficheros `CSV` que contienen todas las métricas junto con descripciones legibles. Este proceso documenta la ejecución para su posterior análisis.

La combinación de estas etapas permite evaluar tanto la calidad geométrica de la alineación como el avance real de la impresión, integrando las métricas necesarias para validar el desempeño del sistema.

A continuación se presenta en la Tabla 4.5 las operaciones principales del módulo de evaluación y comparación, organizadas según la función de cada componente dentro del flujo descrito previamente.

Tabla 4.5 Resumen de operaciones del módulo de evaluación y comparación: etapas, responsabilidades y criterios aplicados.

Componente	Etapas	Responsabilidades principales	Ecuaciones / criterios
Comparator	Evaluación de métodos de registro	Aplica métricas de convergencia a cada transformación propuesta y construye <code>MethodResult</code> .	NN (4.33)–(4.34); RMSE (4.35); overlap (4.36).
<code>convergence_metrics</code>	Cálculo de convergencia geométrica	Computa distancias punto-a-punto, error global y solapamiento bajo tolerancia.	(4.33), (4.34), (4.35), (4.36).
<code>SurfaceCoverageProgress</code>	Progreso de impresión	Cobertura superficial del modelo, completitud de la nube generada, progreso en altura y avance máximo en z .	Cobertura (4.38); completitud (4.40); altura de la nube generada, (4.41)–(4.42); z -avance (4.43).
<code>YZPlotter</code>	Solapamiento lateral y visualización	Proyección YZ , solapamiento proyectado y generación de figura lateral.	Proyección (4.44); solapamiento YZ (4.45).
<code>MethodResult</code>	Encapsulación de resultados	Almacena transformación, nube alineada, métricas, nombre del método y tiempo de ejecución.	No aplica.
<code>MetricsExporter</code>	Exportación de resultados	Genera archivos CSV con las métricas finales y descripciones asociadas.	Criterios de documentación.

Nota: las métricas se reportan habitualmente en porcentaje multiplicando por 100 cuando corresponda.

4.6. Visualización

El módulo de visualización tiene como objetivo generar representaciones bidimensionales para inspeccionar de forma intuitiva la relación geométrica entre las nubes de puntos involucradas en el proceso de registro y evaluación. A partir de las nubes preprocesadas y alineadas en el pipeline, este módulo produce proyecciones en el plano YZ que facilitan la identificación visual de solapes, discrepancias en altura y desviaciones laterales entre la nube generada y el modelo teórico.

Dado que la observación directa en 3D puede dificultar la interpretación, la proyección sobre el plano YZ proporciona una vista lateral para analizar la progresión de la pieza en altura y el grado de coincidencia entre ambos contornos. Además, el módulo incorpora mecanismos de coloreado diferencial e identificación explícita de regiones de intersección, lo que permite generar figuras que complementan las métricas obtenidas en el módulo de evaluación.

Este módulo es utilizado tanto dentro del pipeline como de manera independiente para la inspección manual y posterior documentación de los resultados.

4.6.1. Fundamentos teóricos

La visualización mediante proyección en el plano YZ se realiza para reducir la dimensionalidad de la nube de puntos sin perder información relevante para el análisis de la altura y el perfil lateral de la pieza. Dado un punto tridimensional. [43]

$$\mathbf{x} = (X, Y, Z)^\top$$

Su proyección ortogonal sobre el plano YZ se define como:

$$\pi_{YZ}(\mathbf{x}) = \begin{bmatrix} Y \\ Z \end{bmatrix}. \quad (4.46)$$

La operación consiste simplemente en descartar la coordenada X . Esta proyección permite evaluar diferencias en altura, desplazamientos laterales y cambios en el perfil geométrico de la pieza entre la nube generada y el modelo teórico.

Para cuantificar el solapamiento entre ambas nubes proyectadas, el módulo emplea un criterio geométrico basado en la vecindad en el plano YZ . Considerando las proyecciones $\{(Y_i^A, Z_i^A)\}$ de la nube A y $\{(Y_j^B, Z_j^B)\}$ de la nube B , un punto de B se considera coincidente con A si existe al menos un punto de A situado a una distancia inferior a un umbral

ε . [27]

$$\|(Y_j^B - Y_i^A, Z_j^B - Z_i^A)\| \leq \varepsilon. \quad (4.47)$$

Este criterio se implementa utilizando un árbol de búsqueda espacial, permitiendo determinar rápidamente los puntos intersecantes. A partir de estas coincidencias se calcula un porcentaje de solapamiento. [44]

$$\text{overlap}_{YZ} = \frac{|\{j \mid \exists i \text{ tal que } \|(Y_j^B - Y_i^A, Z_j^B - Z_i^A)\| \leq \varepsilon\}|}{|\{j\}|} \times 100 \quad (4.48)$$

Que expresa qué proporción de los puntos de la nube B se encuentra dentro de la vecindad definida por la nube A en la proyección seleccionada.

Además de la detección de intersecciones, la visualización colorea de forma diferenciada los puntos proyectados, asignando colores distintos para cada nube y un color mezclado para las regiones donde se produce solapamiento geométrico. Esta estrategia facilita la identificación visual inmediata de áreas bien alineadas y zonas con discrepancias.

Finalmente, para garantizar la manejabilidad y legibilidad de las imágenes exportadas, el módulo incluye un mecanismo de submuestreo uniforme que limita la cantidad de puntos representados sin afectar al cálculo del solapamiento, que siempre se realiza sobre la nube completa. Este enfoque permite producir figuras de alta resolución sin comprometer el rendimiento ni la precisión de la métrica visual.

4.6.2. Arquitectura software del módulo

El módulo de visualización está compuesto únicamente por la clase `YZPlotter`; esta centraliza la generación de proyecciones y la superposición en el plano YZ . Tal como se muestra en el diagrama UML de la Figura A.7, esta clase encapsula las operaciones de proyección, cálculo de zonas de intersección y renderizado, permitiendo tanto la exportación a disco como la obtención de imágenes en memoria para usos posteriores en el pipeline.

En términos funcionales, `YZPlotter` implementa tres bloques principales: utilidades internas para proyectar las nubes sobre el plano YZ y gestionar el submuestreo, métodos de superposición y coloreado diferencial entre dos nubes, incluyendo el cálculo de regiones de intersección mediante un árbol KD bidimensional, y métodos de renderizado y exportación de imágenes en formato JPG o como datos codificados en memoria. Esta arquitectura permite integrar el componente de visualización tanto en procesos automáticos del pipeline como en la inspección manual de resultados para documentación o análisis.

Flujo de ejecución

El funcionamiento interno del módulo de visualización sigue una secuencia que permite generar proyecciones informativas y representativas del estado de la impresión y registro. El flujo puede describirse mediante las siguientes etapas:

1. **Proyección de la nube sobre el plano YZ .** A partir de las coordenadas tridimensionales de la nube, `YZPlotter` extrae las componentes (Y, Z) utilizando el método interno `_yz`, que constituye la base de todas las visualizaciones generadas.
2. **Submuestreo uniforme para visualización.** Con el fin de mantener un tamaño de imagen manejable, el módulo aplica un submuestreo coherente mediante `_maybe_subsample`, que selecciona un subconjunto representativo de puntos sin alterar el cálculo del solapamiento, realizado siempre sobre las nubes completas.
3. **Cálculo de zonas de intersección entre dos nubes.** Cuando se comparan dos nubes, el método `_overlap_masks_o3d` determina qué puntos se encuentran dentro de un radio ε en el plano YZ , utilizando un Árbol k -dimensional (`KDTree`) para búsquedas eficientes en la vecindad.
4. **Asignación de colores y mezcla en zonas de solapamiento.** Los puntos pertenecientes a cada nube se colorean de forma diferenciada. En caso de intersección, la mezcla de colores se realiza mediante `_blend_colors`, generando un color distintivo que resalta las regiones coincidentes.
5. **Renderizado y exportación.** Finalmente, con `Matplotlib` se genera la figura resultante y se exporta con `save_pair`, `save_pair_overlap` o `render_pair_overlap`, permitiendo guardar la visualización en disco o integrarla como imagen codificada en memoria dentro del pipeline.

Este flujo proporciona una herramienta visual para evaluar el contorno lateral de la pieza, la coincidencia entre nubes y la calidad geométrica del registro. En la Figura 4.14 se puede observar una demostración de cómo se ven estas proyecciones.

A continuación se resume en la Tabla 4.6 el conjunto de operaciones y responsabilidades del módulo de visualización. Esta tabla complementa el diagrama UML de la Figura A.7 y permite identificar de forma clara qué funciones aporta `YZPlotter` dentro del sistema, así como los criterios geométricos utilizados en la proyección, el cálculo de solapamiento y la generación de representaciones bidimensionales para análisis cualitativo.

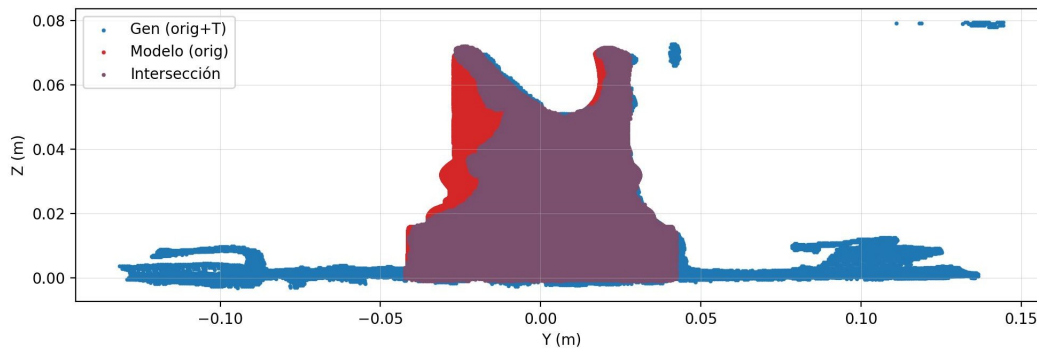


Figura 4.14 Visualización de proyecciones

Tabla 4.6 Resumen de operaciones del módulo de visualización: etapas, responsabilidades y criterios aplicados.

Componente	Etapas	Responsabilidades principales	Ecuaciones / criterios
<code>._yz</code>	Proyección 2D	Obtiene la proyección ortogonal (Y, Z) de cada punto de la nube.	Proyección (4.46).
<code>._maybe_subsample</code>	Gestión de densidad visual	Reduce la cantidad de puntos para visualizar sin alterar métricas reales.	Submuestreo uniforme.
<code>._overlap_masks_o3d</code>	Cálculo de solapamiento	Determina qué puntos se encuentran a distancia $\leq \varepsilon$ en el plano YZ .	Condición de vecindad (4.47), porcentaje (4.48).
<code>._blend_colors</code>	Visualización de intersecciones	Genera un color mezclado para representar regiones de solapamiento entre nubes.	Mezcla lineal de colores.
<code>save_pair_overlap</code>	Superposición y exportación	Superpone dos nubes, colorea intersecciones y genera imágenes en disco o en memoria.	Porcentaje de solapamiento (4.48).
<code>save_single</code>	Visualización individual	Genera la proyección YZ de una única nube para inspección lateral.	Proyección (4.46).

4.7. Interfaz gráfica de usuario

La interfaz gráfica tiene como objetivo facilitar el uso del sistema sin necesidad de interactuar directamente con la línea de comandos. A través de un conjunto de ventanas, se dirige al usuario por todo el flujo de trabajo.

El alcance de la interfaz se centra en dos operaciones principales: la ejecución completa del proceso de validación sobre una adquisición y la inspección de nubes de puntos almacenadas.

4.7.1. Flujos de uso principales

Ejecución del proceso completo de validación

El primer flujo de uso corresponde a la ejecución del proceso de validación sobre un modelo teórico seleccionado por el usuario. Desde el menú principal, el usuario accede a la ventana de configuración del proceso, donde se carga el archivo que representa el modelo .stl de la pieza y se especifica el tamaño de vóxel con el que se discretizarán las nubes de puntos.

Una vez definidos estos parámetros, la interfaz permite lanzar la ejecución del sistema. En segundo plano, se activa el módulo de adquisición para capturar la escena con la cámara RGB-D, se genera la nube de puntos correspondiente y se aplica la cadena de preprocesado. A continuación, se lleva a cabo el registro entre la nube generada y el modelo teórico y se calculan las métricas de validación asociadas al progreso de impresión y a la calidad del alineamiento.

Al finalizar el proceso, la Interfaz Gráfica de Usuario (GUI) presenta al usuario un resumen de los resultados en forma de tabla con los valores de las métricas más relevantes, junto con una proyección en el plano YZ. De este modo, el flujo de trabajo queda encapsulado en una secuencia guiada.

Visualización de nubes de puntos existentes

El segundo flujo de uso está orientado a la inspección de nubes de puntos previamente generadas. Desde el menú principal, el usuario puede acceder a la ventana de visualización, donde se ofrece la posibilidad de seleccionar un archivo individual o una carpeta que contenga múltiples nubes de puntos en formato .PLY.

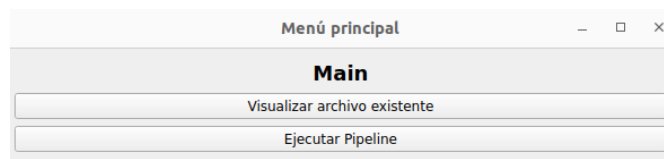
Tras la selección, la interfaz permite cargar y representar las nubes tridimensionalmente o sus proyecciones sobre los planos.

4.7.2. Detalles de implementación

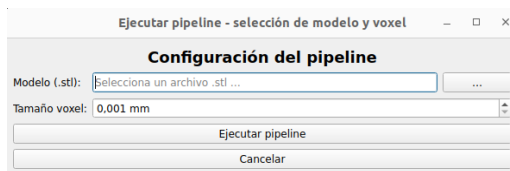
La interfaz gráfica se ha implementado en Python siguiendo un modelo basado en ventanas, donde cada funcionalidad principal se encapsula en una clase distinta. El archivo `main.py` actúa como punto de entrada de la aplicación y se encarga de inicializar el entorno gráfico y mostrar la ventana de menú principal. Esta ventana se implementa en `MenuWindow.py` y ofrece las dos rutas de trabajo descritas anteriormente: ejecución del proceso completo y visualización de nubes existentes.

Las ventanas asociadas a la configuración se definen en los ficheros `Window.py`. La lógica común de gestión de rutas, diálogos de selección de archivos y mensajes de estado se agrupa en `utils_window.py`. Cada ventana se limita a recoger las entradas del usuario y a mostrar resultados, delegando las operaciones de adquisición, preprocesado, registro y evaluación en los módulos específicos del sistema.

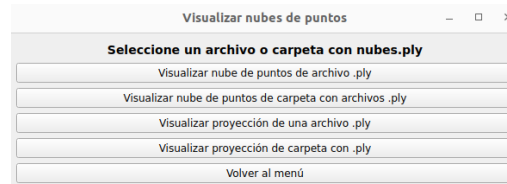
Esta organización favorece la separación entre la lógica de presentación y la lógica del proceso. En la Figura 4.15 se puede observar un diagrama de navegación dentro de las ventanas.



(a) Pantalla Principal

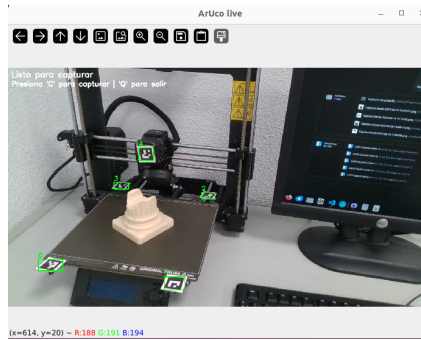


(b) Ejecutar Pipeline

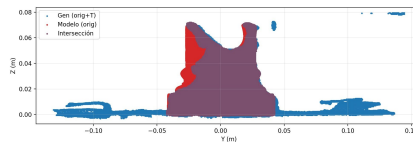


(c) Visualizar nubes existentes

Figura 4.15 Interfaz gráfica del sistema.



(a) Segmentación de los ArUco



(b) Proyección en el plano YZ

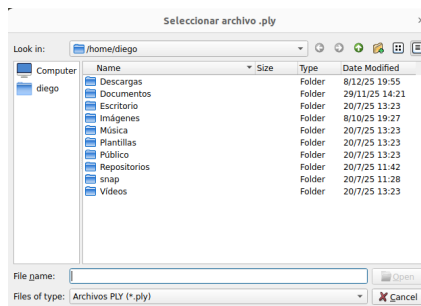
Resultados de registro

Resultados de los métodos de registro

Método	Seguimiento	Coverage	Porcentaje de impresión por objeto	Porcentaje de impresión por superficie	Porcentaje de impresión por volumen
Y-Método: general	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %
Z-Método: +O2 (puntos por plano)	10.00 %	10.00 %	10.00 %	10.00 %	10.00 %

(c) Tabla de resultados

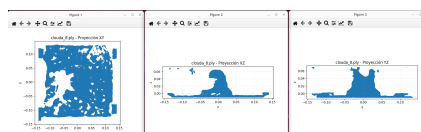
Figura 4.16 Flujo de trabajo para la interfaz de la Figura 4.15b



(a) Seleccionar archivo a visualizar



(b) Representación de las nubes de puntos



(c) Proyecciones en los planos de la nube de puntos

Figura 4.17 Flujo de trabajo para la interfaz de la Figura 4.15c

Resultados y discusión

En este capítulo, se analizan los resultados obtenidos experimentalmente. En particular, se estudia la robustez del proceso planteado para diferentes niveles de progresión de la pieza, la estabilidad del proceso frente a distintas configuraciones de captura y la coherencia de las métricas para validar la impresión.

5.1. Diseño de los experimentos y conjunto de datos

Para evaluar el desempeño del algoritmo propuesto, se diseñó una metodología experimental con distintas fases de progresión de la pieza y configuraciones de captura. Se busca someter al sistema a escenarios representativos del proceso real de impresión 3D, donde la geometría observada evoluciona.

Los experimentos se realizaron sobre una pieza que presenta tres zonas diferenciadas en altura. Esta se emplea a lo largo de todo el capítulo como referencia para interpretar los resultados:

- **Nivel 1: sección baja.** Corresponde a las primeras capas de impresión, donde únicamente se ha materializado la parte inferior de la pieza. La nube generada contiene un 19 % de la geometría total. Figura 5.1
- **Nivel 2: sección baja + media.** Incluye tanto la base como la zona intermedia, proporcionando una geometría con 52 % de la pieza. Figura 5.2
- **Nivel 3: sección completa.** Representa el estado total de impresión de la pieza. Figura 5.3

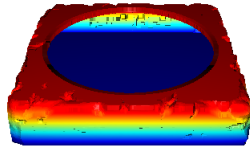


Figura 5.1 Nivel 1: Sección baja de la pieza impresa

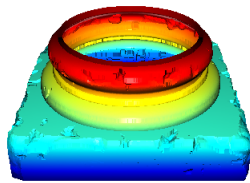


Figura 5.2 Nivel 2: Sección baja + media de la pieza impresa

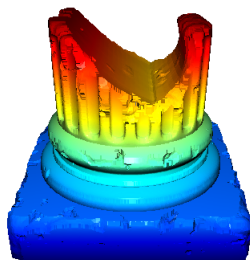


Figura 5.3 Nivel 3: Sección completa de la pieza impresa

5.1.1. Escenarios de progresión y poses de cámara

Para cada uno de los tres niveles de progresión, la cámara RGB-D se colocó en dos posiciones distintas respecto a la pieza y se tuvo en cuenta que los 5 marcadores ArUco estuvieran visibles. Por otro lado, estas dos poses presentan ángulos de observación diferentes, de modo que cada una captura zonas distintas de la pieza. En las Figuras 5.4 y 5.5 se pueden observar las diferentes posiciones usadas.

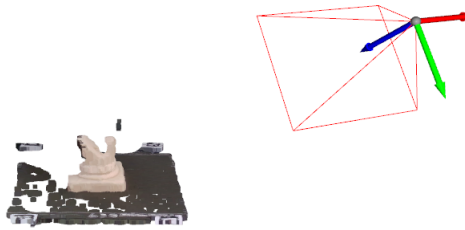


Figura 5.4 Adquisición en la pose 1 de la cámara.

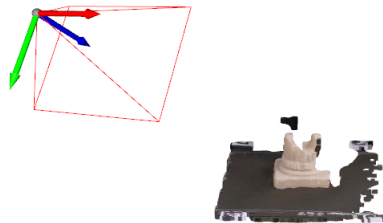


Figura 5.5 Adquisición en la pose 2 de la cámara.

En cada nivel de progresión se adquirieron 200 capturas para cada una de las poses. En total, cada nivel de progresión cuenta con 400 adquisiciones individuales, que permiten estudiar la variabilidad del registro y de las métricas. Además, se construyó un escenario fusionado, combinando nubes obtenidas desde ambas poses. Esto se hizo para obtener una reconstrucción más completa de la pieza y someter al algoritmo a un caso más exigente, con mayor cantidad de puntos y más ruido acumulado. Una demostración de esto se encuentra en la Figura 5.6



Figura 5.6 Adquisición fusionada.

5.2. Desempeño del algoritmo en el registro de nubes de puntos

El solapamiento, *overlap*, se calcula como el porcentaje de puntos de la nube generada cuya distancia NN al modelo es menor que una tolerancia τ , según (4.36). De forma complementaria, el error global del alineamiento puede resumirse con $RMSE_{sym}$ en (4.35).

5.2.1. Visión global del registro para una pose

En las Figuras 5.7, 5.8 y 5.9 se muestra la distribución del solapamiento obtenido en el registro para una única pose de cámara y los seis métodos de registro considerados. Cada diagrama de caja resume la información de 400 capturas en cada nivel.

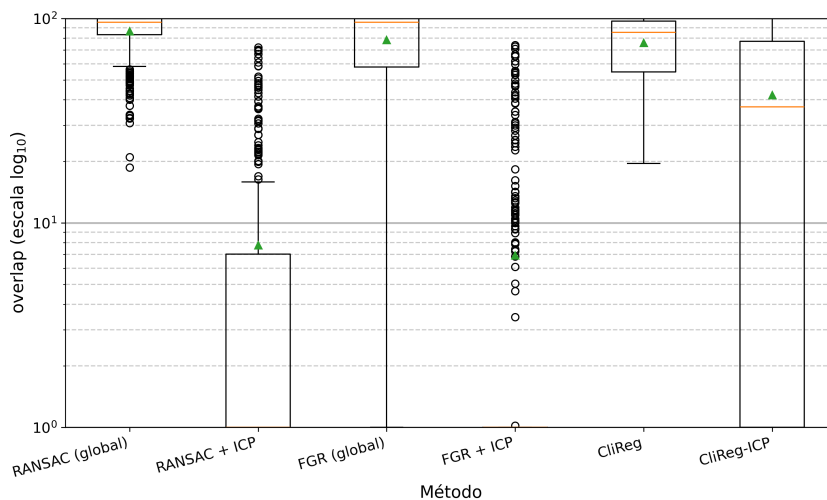


Figura 5.7 Solapamiento para el registro del nivel bajo de la pieza.

En estas figuras se observa que el solapamiento entre la nube generada y el modelo teórico mejora a medida que aumenta la altura de la pieza y, por tanto, la cantidad de

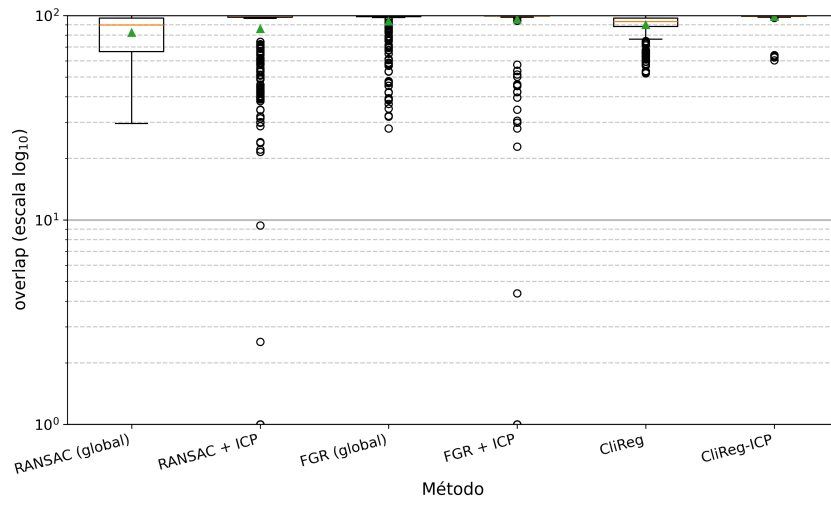


Figura 5.8 Solapamiento para el registro del nivel bajo y medio de la pieza.

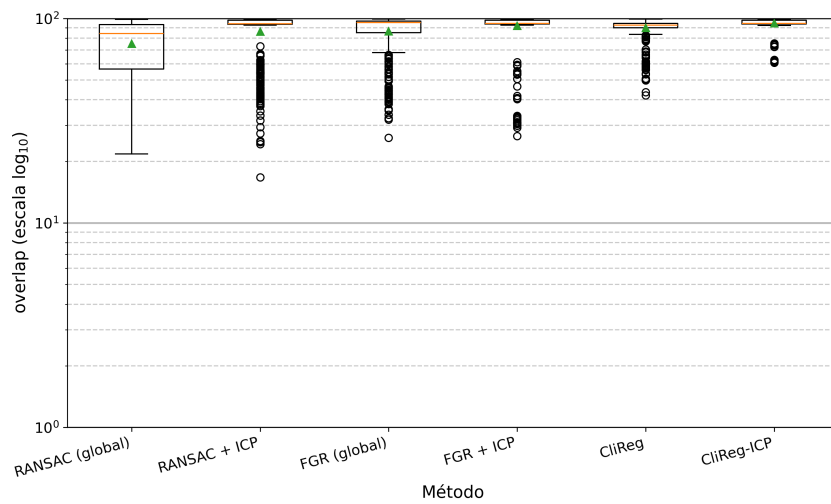


Figura 5.9 Solapamiento para el registro de la pieza completa.

geometría disponible para el registro. En la sección baja, Figura 5.7, las distribuciones son mucho más dispersas y aparecen valores muy bajos, especialmente en los métodos que incluyen un refinamiento local con ICP. En cambio, para los niveles consecutivos, Figuras 5.8 y 5.9, todas las cajas se concentran cerca de valores altos de solapamiento, con medianas en torno al 90–100 % según el método. Esto indica que el proceso diseñado es capaz de alinear de forma consistente la nube generada con el modelo cuando la impresión está razonablemente avanzada.

La escala logarítmica del eje vertical permite apreciar mejor las diferencias entre fallos graves de registro, valores próximos al 1 %, y alineaciones satisfactorias, por encima del 80–90 %. En la sección baja, las cajas de algunos métodos se extienden desde la parte inferior de la gráfica hasta valores moderados de solapamiento, reflejando una alta sensibilidad al ruido y la geometría parcial. Por el contrario, en los niveles intermedio y completo, las cajas son más compactas y los bigotes se acortan, lo que sugiere una mayor estabilidad del sistema frente a variaciones entre capturas individuales.

En conjunto, la robustez del registro está condicionada por la cantidad de información geométrica disponible. En las primeras capas, la pieza presenta poca altura y superficie observable, lo amplifica el impacto del ruido del sensor y favorece mínimos locales, resultando en variaciones elevadas y divergencia. Este comportamiento es especialmente relevante en FDM, donde los defectos críticos suelen manifestarse al inicio del proceso. Sin embargo, a partir del nivel intermedio, el incremento de superficie impresa proporciona suficientes restricciones geométricas para que el registro tienda a converger de manera consistente, con distribuciones más compactas y solapamientos altos. Esta transición justifica que el método propuesto sea especialmente robusto y reproducible para niveles de impresión medios/altos, mientras que su capacidad para identificar fallos en etapas tempranas queda limitada, aspecto que se retoma explícitamente en el capítulo de conclusiones y trabajo futuro.

5.2.2. Comparación por nivel de progresión

Nivel 1: sección baja

En la sección baja de la pieza, Figura 5.7, los métodos globales RANSAC y FGR alcanzan medianas de solapamiento cercanas al 95 %, lo que indica que, cuando consiguen encontrar una solución coherente, son capaces de alinear razonablemente bien la nube parcial con el modelo. Sin embargo, la desviación es elevada y aparecen valores dispersos, lo que revela una falta de robustez. Esto indica que pequeñas variaciones en la adquisición o en la distribución de puntos pueden conducir a soluciones diferentes.

La situación es más evidente en los métodos que combinan un registro global con un

refinamiento local ICP. Tanto RANSAC+ICP como FGR+ICP presentan medianas en torno al 1 %, con gran parte de las muestras concentradas en el extremo inferior de la gráfica. Esto indica que, en una fracción importante de las capturas, el refinamiento local no corrige la solución inicial sino que la degrada, quedando atrapado en mínimos locales debidos a la escasa geometría disponible y a la presencia de ruido. En otras palabras, con una pieza muy baja aplicar el refinamiento por ICP no resulta fiable.

Los métodos basados en CliReg muestran un comportamiento intermedio. La versión sin refinamiento alcanza una mediana cercana al 85 %, con menor concentración de valores extremos, mientras que CliReg-ICP presenta una mediana en torno al 40 % y una amplia dispersión. Estos resultados sugieren que el método aporta cierta estabilidad frente al solapamiento parcial, pero que el refinamiento ICP sigue degradando el resultado.

En conjunto, los resultados de la sección baja apuntan a que existe una altura mínima a partir de la cual las métricas de solapamiento pueden interpretarse con confianza. Por debajo de ese umbral, la combinación de pocas capas impresas, ruido del sensor y oclusiones hace que el registro sea muy sensible a perturbaciones.

Nivel 2: sección baja+media

Cuando la impresión incluye tanto la sección baja como la intermedia, Figura 5.8, la distribución del solapamiento mejora para todos los métodos. Las medianas se sitúan entre el 89 % y el 100 % según el algoritmo, y las cajas se comprimen hacia la parte superior de la gráfica.

En este escenario los métodos con refinamiento ICP dejan de empeorar el resultado. RANSAC+ICP y FGR+ICP muestran medianas en torno al 99 %, con cajas muy estrechas y bigotes cortos, lo que indica que la solución final es casi siempre una alineación muy precisa. De forma similar, CliReg-ICP presenta una mediana también cercana al 99 %, manteniendo una variabilidad reducida. La presencia de una geometría con más altura y rasgos característicos provoca que el refinamiento local pueda ajustar la transformación.

Los métodos puramente globales mantienen también un buen comportamiento: FGR alcanza medianas próximas al 99 %, mientras que RANSAC se sitúa algo por debajo, alrededor del 90 %, pero aun así con un solapamiento claramente alto y estable. CliReg sin ICP obtiene una mediana en torno al 93 %, ligeramente inferior a las variantes con refinamiento, pero con una desviación moderada. Esta mejora puede interpretarse como consecuencia a disponer de más superficie a analizar.

Nivel 3: pieza completa

En el nivel más avanzado, donde la pieza incluye las secciones baja, media y alta, Figura 5.9, el algoritmo se estabiliza y la mayoría de métodos alcanzan solapamientos elevados de forma consistente. Las medianas de RANSAC+ICP, FGR, FGR+ICP, CliReg y CliReg-ICP se sitúan en 100 %, con cajas estrechas y una variabilidad baja. El método RANSAC presenta una mediana algo menor, alrededor del 84 %, pero mantiene igualmente un comportamiento aceptable.

Al completar la pieza se obtiene una mejora de estabilidad y una reducción de la variación. Esto indica que al alcanzar una altura intermedia, añadir capas refuerza la redundancia geométrica pero no cambia la dificultad del problema de registro.

Desde el punto de vista de la validación de la impresión, estos resultados indican que, para una única pose de cámara, algoritmo ofrece métricas de solapamiento fiables a partir del momento en que se ha impreso al menos la combinación baja+media.

5.2.3. Visión global del registro para dos poses fusionadas

Además de evaluar el registro a partir de capturas individuales, se analizó el comportamiento del algoritmo cuando la nube generada se obtiene como fusión de dos poses de cámara. Las Figuras 5.10, 5.11 y 5.12 recogen la distribución del solapamiento en este escenario para los tres niveles de progresión de la pieza.

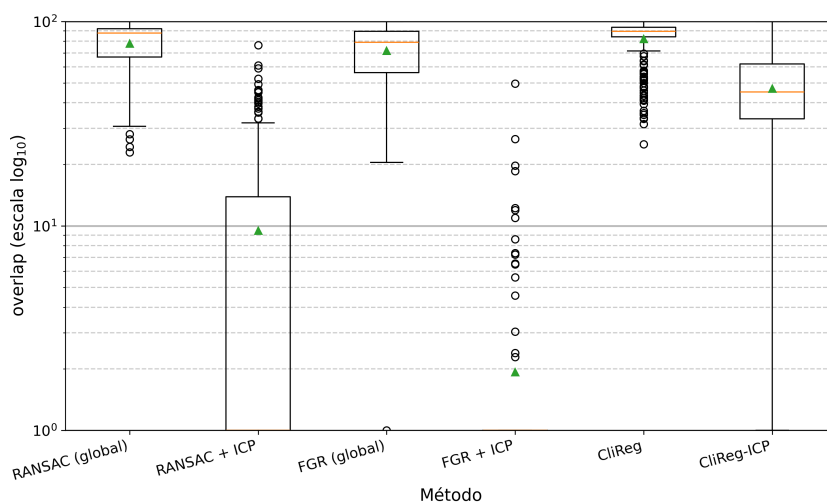


Figura 5.10 Solapamiento para el registro del nivel bajo con captura de dos poses diferentes.

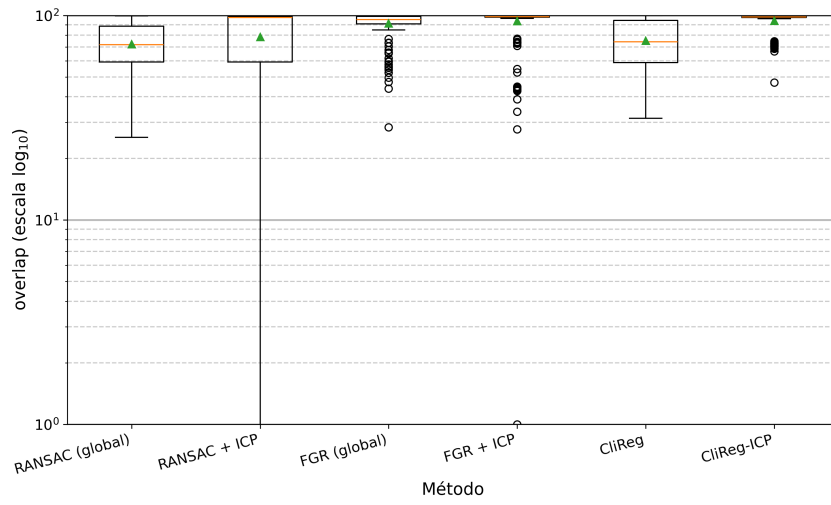


Figura 5.11 Solapamiento para el registro del nivel bajo+medio con captura de dos poses diferentes.

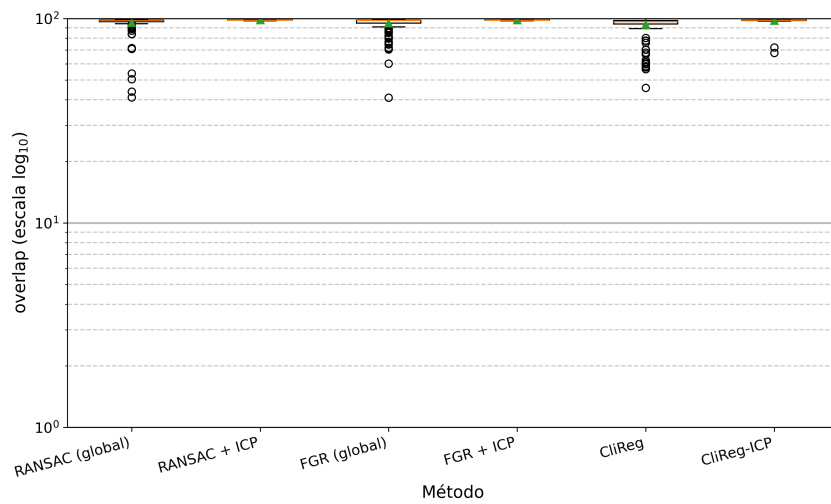


Figura 5.12 Solapamiento para el registro de la pieza completa con captura de dos poses diferentes.

Nivel 1: sección baja

En el nivel bajo, Figura 5.10, la fusión de poses mejora el comportamiento de algunos métodos, pero no resuelve la falta de robustez observada en la sección anterior. Los métodos globales, RANSAC y FGR, y CliReg alcanzan ahora medianas de solapamiento en torno al 80–90 %, con cajas relativamente compactas y unos cuantos casos de registro fallido. En comparación con la captura de una única pose, la fusión proporciona más geometría y reduce ligeramente la presencia de outliers extremos en CliReg, pero también hace que RANSAC y FGR sean algo más sensibles a variaciones locales, desplazando sus medianas ligeramente hacia abajo.

Los métodos que combinan un registro global con ICP siguen mostrando un comportamiento deficiente. RANSAC+ICP y FGR+ICP concentran la mayor parte de las muestras en valores de solapamiento bajos, lo que indica que el refinamiento local continúa divergiendo en la mayor parte de las capturas. Aunque la fusión de poses aporta más puntos, la geometría disponible sigue siendo demasiado pobre y ruidosa para que el refinamiento encuentre una solución. En este nivel, tanto con una pose como con dos, el registro de la pieza resulta inestable y las métricas no son fiables para validar la impresión.

Nivel 2: sección baja+media

Cuando se fusionan dos poses en el nivel bajo+medio, Figura 5.11, la mayor parte de los métodos alcanzan solapamientos altos, típicamente por encima del 90 %.

FGR y CliReg-ICP se sitúan entre los métodos más robustos; sus medianas se mantienen muy cercanas al 100 % y las cajas son estrechas, lo que indica que, para la mayoría de capturas, la nube fusionada se alinea casi por completo con el modelo. RANSAC+ICP y FGR+ICP en muchas ejecuciones alcanzan también solapamientos muy altos, pero aparecen algunos fallos de registro que no estaban presentes en el caso de una única pose. Esto sugiere que el ICP se beneficia de la información adicional en la mayoría de situaciones, pero puede verse penalizado cuando las dos vistas presentan discrepancias o zonas mal reconstruidas.

CliReg sin refinamiento mantiene solapamientos elevados pero ligeramente inferiores a su variante con ICP, mientras que RANSAC se sitúa en un rango intermedio, medianas en torno al 70–80 % y mayor dispersión. En conjunto, la fusión de poses refuerza el comportamiento de los métodos más robustos, FGR y CliReg-ICP, pero introduce cierta variabilidad adicional en los algoritmos que dependen fuertemente de un refinamiento local.

Nivel 3: pieza completa

En el nivel completo, Figura 5.12, la combinación de una geometría completa con la fusión de dos poses hace que todos los métodos con refinamiento alcancen solapamientos muy altos que se sitúan prácticamente en el 100 %, con cajas muy comprimidas y pocos outliers. RANSAC, FGR y CliReg presentan también medianas elevadas, por encima del 99 %, aunque con una mayor dispersión.

Comparado con el caso de una pose, la nube fusionada mantiene o mejora el nivel de solapamiento medio y refuerza la confianza en los métodos basados en FGR y en CliReg-ICP, que se comportan de forma estable en prácticamente todas las adquisiciones. La presencia residual de outliers sugiere que aún pueden aparecer configuraciones desfavorables, pero los resultados respaldan la utilización del caso fusionado como escenario de validación exigente para el pipeline.

5.3. Métricas de progreso de impresión

5.3.1. Progreso en altura: `zmax_percent`

La métrica `zmax_percent` estima el avance de la impresión a partir de la altura máxima de la nube de puntos registrada, expresada como porcentaje de la altura total del modelo teórico. De este modo, para los niveles de progresión definidos, la sección baja cuenta con un 19 % de impresión, baja+media 52 % y pieza completa 100 %. Se espera que la distribución de `zmax_percent` se concentre en torno a esos valores cuando el registro es correcto.

Esta métrica se define formalmente en (4.43), y constituye un estimador de avance basado en la componente máxima z de la nube registrada en relación con la altura total del modelo.

En las Figuras 5.13, 5.14 y 5.15 se representa la distribución de `zmax_percent` obtenida a partir de una única pose de cámara para los seis métodos de registro evaluados. Por otro lado, las Figuras 5.16, 5.17 y 5.18 muestran el comportamiento de la métrica cuando la nube generada se obtiene fusionando dos poses distintas.

Nivel 1: sección baja

En el nivel bajo, Figuras 5.13 y 5.16, las medianas de `zmax_percent` para los métodos RANSAC, FGR y CliReg se sitúan en torno a la altura teórica esperada, con valores típicos entre el 15 % y el 25 %. Esto indica que, cuando el registro es razonablemente

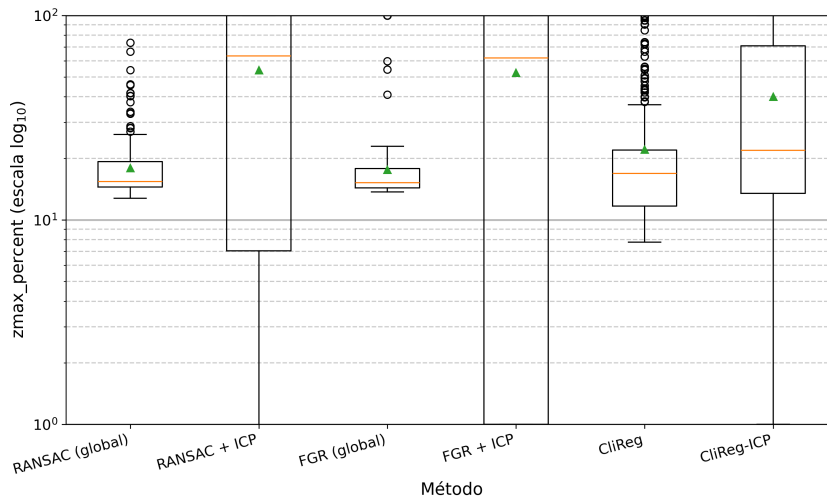


Figura 5.13 Distribución de `zmax_percent` para el registro del nivel bajo de la pieza con una única pose de cámara.

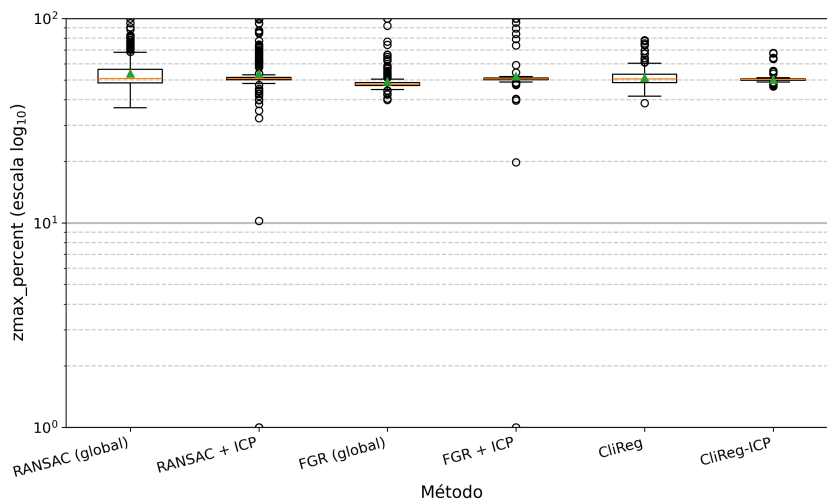


Figura 5.14 Distribución de `zmax_percent` para el registro del nivel bajo+medio de la pieza con una única pose de cámara.

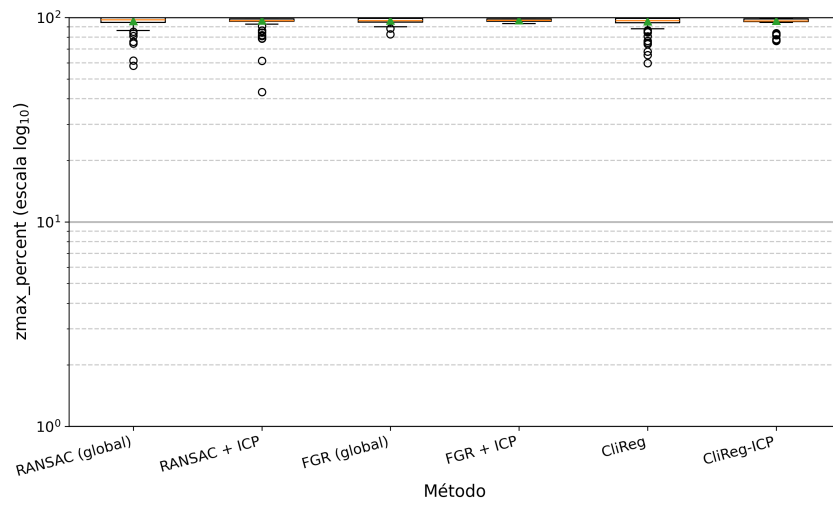


Figura 5.15 Distribución de `zmax_percent` para el registro de la pieza completa con una única pose de cámara.

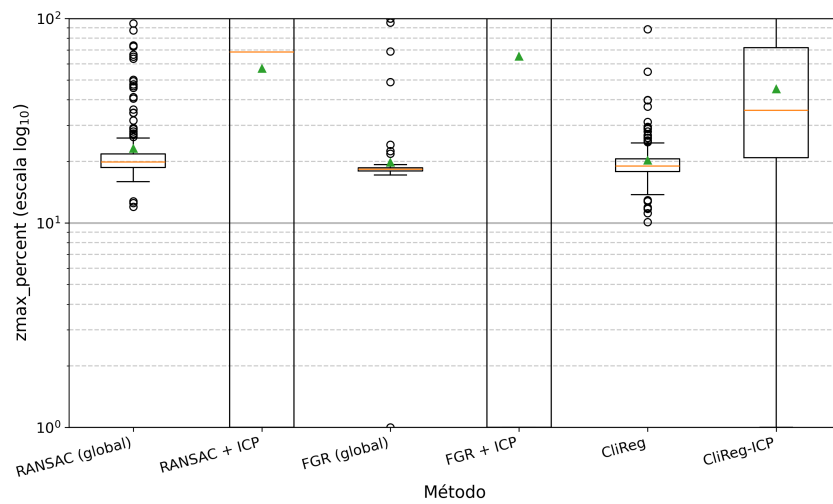


Figura 5.16 Distribución de `zmax_percent` para el registro del nivel bajo a partir de la fusión de dos poses de cámara.

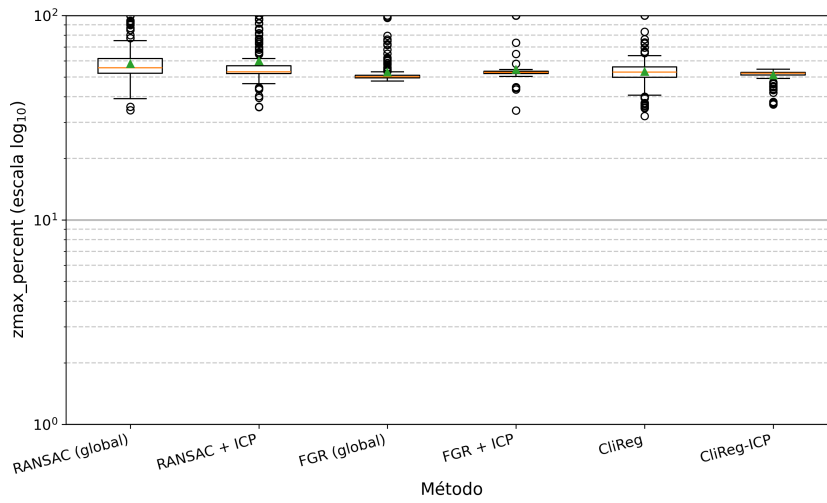


Figura 5.17 Distribución de `zmax_percent` para el registro del nivel bajo+medio a partir de la fusión de dos poses de cámara.

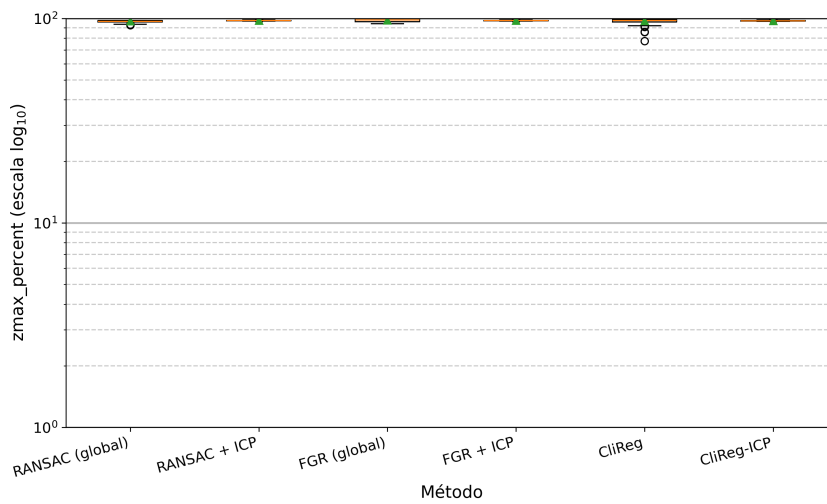


Figura 5.18 Distribución de `zmax_percent` para el registro de la pieza completa a partir de la fusión de dos poses de cámara.

correcto, la métrica refleja de forma coherente que sólo una fracción reducida de la pieza ha sido impresa.

Sin embargo, la dispersión es elevada y aparecen valores atípicos que alcanzan porcentajes muy superiores, 60–70 %, especialmente en los métodos que incorporan un refinamiento local por ICP. Estos casos corresponden a registros fallidos. Desde el punto de vista del progreso, `zmax_percent` interpreta estos errores como una impresión mucho más avanzada de lo que realmente es.

Nivel 2: sección baja+media

Al pasar al nivel intermedio de 52 %, Figuras 5.14 y 5.17, las distribuciones se desplazan de forma consistente hacia valores en torno al 50–60 %, próximos al porcentaje teórico. Las cajas se comprimen y los bigotes se acortan, lo que indica que la altura máxima observada en la nube alineada es mucho más estable entre capturas.

La presencia de una mayor cantidad de geometría hace que los errores de registro sean menos frecuentes y que el recorte basado en el modelo teórico funcione de manera más fiable. Aun así, se observan valores atípicos por debajo del 20 %, que corresponden a registros fallidos en los que la nube queda desplazada significativamente en altura. En este rango, la fusión de dos poses no altera la mediana de `zmax_percent`, pero tiende a reducir ligeramente la variabilidad en los métodos más robustos.

Nivel 3: pieza completa

En el nivel completo, Figuras 5.15 y 5.18, las medianas se sitúan muy próximas al 100 % y las cajas son extremadamente estrechas, especialmente en el caso fusionado. Sólo aparecen algunos atípicos por debajo del 80–90 %, que reflejan situaciones en las que el recorte o el registro dejan la nube ligeramente por debajo de la parte superior del modelo.

En este escenario, `zmax_percent` se comporta como un indicador muy fiable de que la pieza ha alcanzado su altura final, prácticamente independiente del método de registro escogido siempre que éste proporcione una alineación razonable.

En conjunto, los resultados muestran que `zmax_percent` responde de forma coherente al incremento de altura teórica de la pieza. La distribución de la métrica se desplaza desde valores próximos al 19 % hasta valores muy cercanos al 100 % a medida que progresa la impresión, tanto con una única pose como en el caso fusionado. Esto respalda su uso como estimador del avance global de la impresión.

No obstante, al basarse únicamente en la altura máxima de la nube, `zmax_percent` es ciega a la presencia de huecos internos o defectos en las capas inferiores: una nube con

muchas variaciones puede seguir alcanzando la misma altura que una impresión correcta. Además, la métrica es sensible al ruido y a puntos aislados por encima de la superficie real, que pueden hacer que la altura máxima se dispare artificialmente. Por ello, aunque `zmax_percent` resulta muy útil como indicador y computacionalmente ligero del estado de la impresión, su interpretación debe complementarse con métricas geométricas para realizar una validación robusta del proceso.

5.3.2. Cobertura superficial: `progress_surface`

La métrica `progress_surface` comprueba para cada punto de la nube teórica si existe un punto de la nube registrada a una distancia menor que τ . Formalmente, para cada punto del modelo se calcula la distancia a su vecino más cercano en la nube generada (4.37) y se computa la fracción de puntos cubiertos bajo tolerancia τ según (4.38).

Esta métrica se implementa en la clase `SurfaceCoverageProgress`, que opera sobre las nubes ya alineadas y recortadas por las utilidades de preprocesado, de modo que la comparación se realiza sólo en la región del modelo observable por la cámara.

Las Figuras 5.19, 5.20 y 5.21 muestran la distribución de `progress_surface` para una única pose de cámara en los tres niveles de progresión de la pieza. Las Figuras 5.22, 5.23 y 5.24 recogen los resultados cuando la nube generada se obtiene mediante la fusión de dos poses distintas.

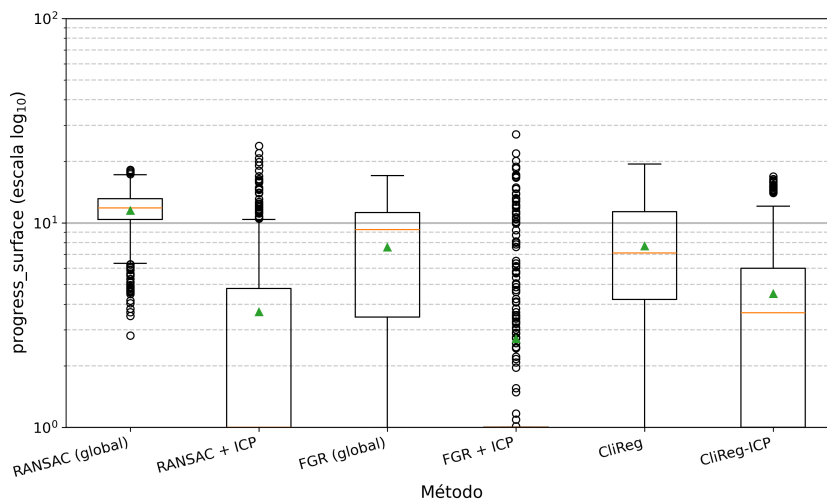


Figura 5.19 Distribución de `progress_surface` para el nivel bajo de la pieza con una única pose de cámara.

En conjunto, las figuras demuestran que `progress_surface` es una métrica más exigente que `zmax_percent`. Mientras que la altura máxima puede coincidir con la teórica aún cuando existan huecos importantes, `progress_surface` sólo crece cuando la nube es capaz de coincidir con la superficie visible del modelo. Esto hace que la métrica sea

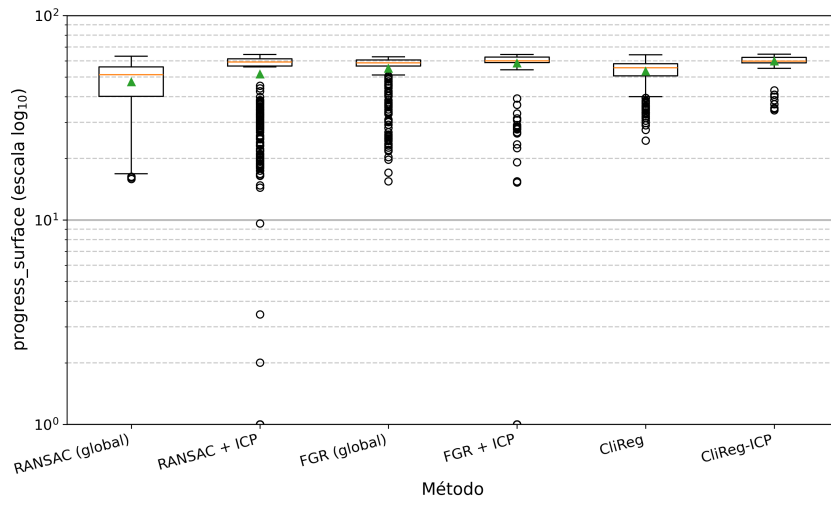


Figura 5.20 Distribución de `progress_surface` para el nivel bajo+medio de la pieza con una única pose de cámara.

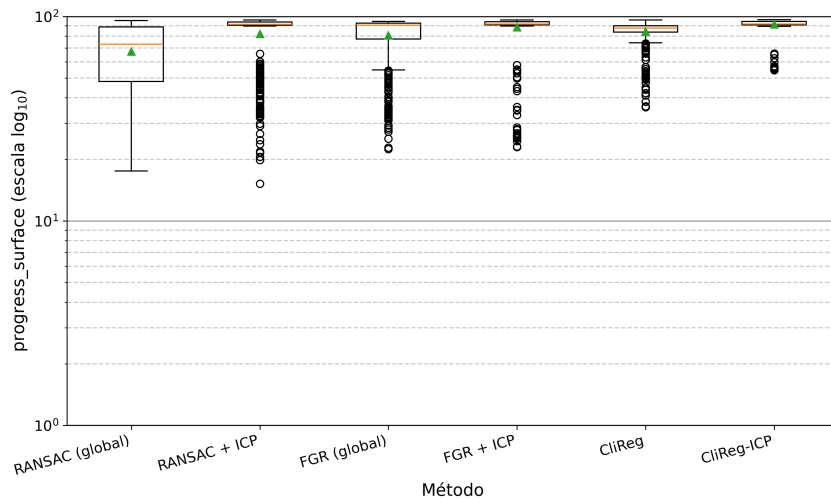


Figura 5.21 Distribución de `progress_surface` para la pieza completa con una única pose de cámara.

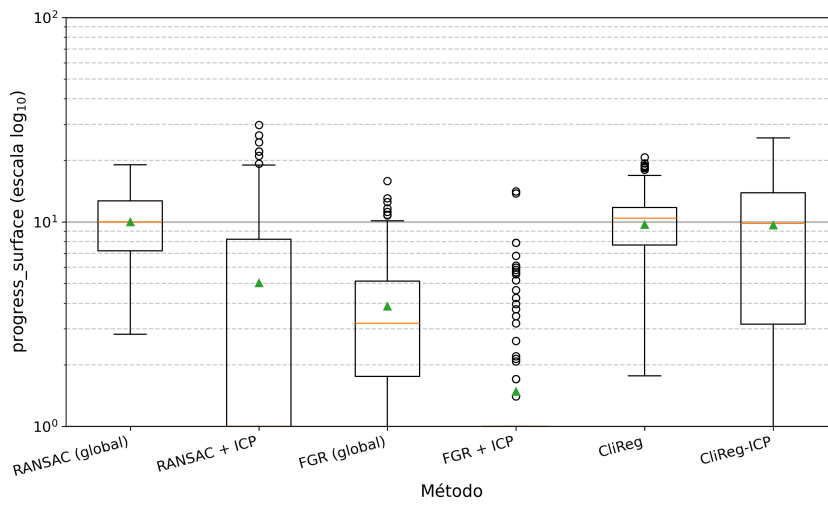


Figura 5.22 Distribución de `progress_surface` para el nivel bajo a partir de la fusión de dos poses de cámara.

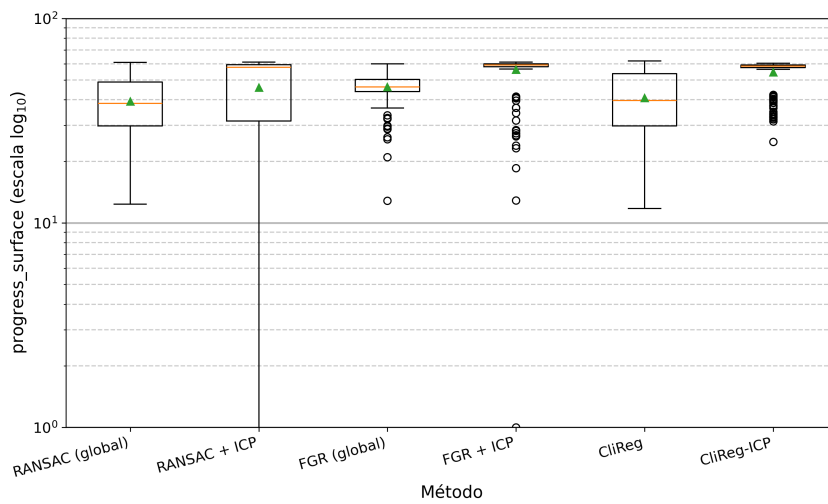


Figura 5.23 Distribución de `progress_surface` para el nivel bajo+medio a partir de la fusión de dos poses de cámara.

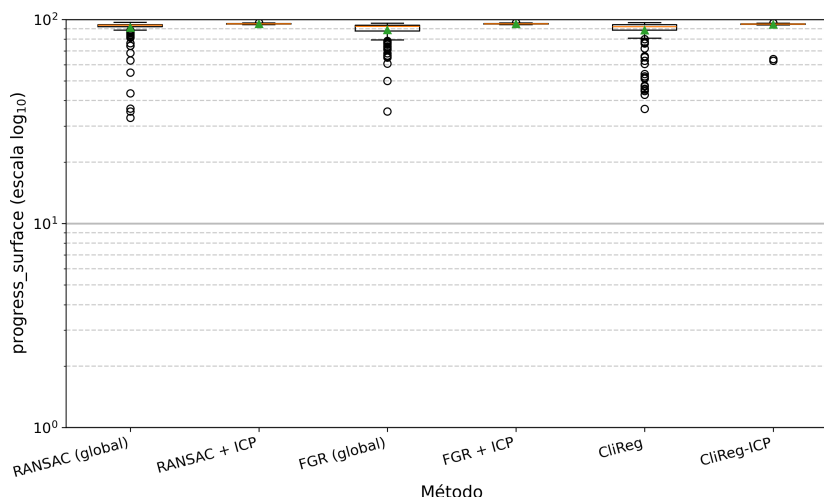


Figura 5.24 Distribución de `progress_surface` para la pieza completa a partir de la fusión de dos poses de cámara.

especialmente sensible a errores de registro y decisiones de recorte, pero también la convierte en un indicador más fino de la calidad geométrica del proceso. A diferencia de `zmax_percent` (4.43), `progress_surface` (4.38) sólo aumenta cuando existe coincidencia local con la superficie teórica bajo τ .

Nivel 1: sección baja

En el nivel bajo de 19 %, Figuras 5.19 y 5.22, las medianas de `progress_surface` con una única pose se sitúan en torno al 10 % para los mejores métodos y por debajo para las variantes más inestables. Esto confirma que, con tan poca altura de pieza, la cámara sólo observa una fracción muy limitada de la superficie teórica y pequeñas desviaciones en el registro se traducen en pérdidas significativas de cobertura. En particular, las combinaciones RANSAC+ICP, FGR+ICP y CliReg+ICP concentran una parte importante de las muestras cerca del 1–5 %, reflejando casos en los que el registro se degrada y la nube prácticamente deja de solapar con el modelo.

La fusión de dos poses incrementa ligeramente la cobertura en algunos métodos, como RANSAC o CliReg, que alcanzan medianas algo superiores y una dispersión más moderada. Sin embargo, para las variantes con ICP la situación no mejora ya que aparecen numerosas capturas con coberturas muy bajas.

Nivel 2: sección baja+media

Cuando la pieza alcanza el 52 % de altura, Figuras 5.20 y 5.23, la cobertura superficial aumenta de forma clara. Con una única pose, las medianas de `progress_surface` se sitúan

en torno al 50–60 %, y las cajas se vuelven más compactas. La mejora está ligada a la aparición de más rasgos geométricos y a que la región de recorte del modelo, definida por las utilidades de visibilidad, abarca una porción mayor de la pieza sin incluir zonas completamente ocultas.

Al fusionar dos poses, los métodos basados en FGR+ICP y CliReg+ICP alcanzan medianas próximas a 57 % y muestran poca variabilidad entre capturas. La segunda vista permite cubrir zonas que en la captura simple quedaban ocultas, de modo que la fracción de superficie coincidente aumenta sin necesidad de modificar la tolerancia τ . Aun así, las variantes con ICP siguen presentando un número apreciable de casos con baja cobertura debido a registros erróneos, lo que ilustra que `progress_surface` penaliza con fuerza las transformaciones incorrectas.

Nivel 3: pieza completa

En el nivel completo, Figuras 5.21 y 5.24, para las capturas con una sola pose, el método RANSAC alcanza una mediana del 80 % y con mayor desviación. En cambio, RANSAC+ICP, FGR, FGR+ICP, CliReg y CliReg+ICP logran medianas que se aproximan al 95 % con una menor desviación.

Cuando se emplea la fusión de dos poses, todas las cajas se desplazan hacia valores cercanos al 96 %, y la variabilidad se reduce de forma notable. En este escenario, `progress_surface` actúa como una métrica muy exigente.

Finalmente, `progress_surface` proporciona una medida de cuánta superficie de la pieza está siendo correctamente reconstruida por la nube registrada. Su comportamiento a lo largo de los niveles de impresión y entre captura simple y fusión de vistas confirma que la métrica es sensible tanto al progreso real de la impresión como a la calidad del registro. Esto la convierte en un complemento de `zmax_percent` para la validación geométrica del proceso.

5.3.3. Coincidencia volumétrica: `perc_b_in_a`

La métrica `perc_b_in_a` cuantifica qué porcentaje de puntos de la nube generada encuentra una pareja en el modelo teórico dentro de una tolerancia espacial τ . Esta variable mide hasta qué punto el volumen ocupado por la nube adquirida puede explicarse por la nube teórica. Valores altos indican que la mayor parte de los puntos observados se encuentran en el interior de la pieza teórica, mientras que valores bajos revelan ruido o zonas mal alineadas. La métrica se define a partir de la distancia del punto generado al modelo (4.39) y se expresa como la fracción de puntos de \mathcal{G} que encuentran correspondencia dentro de τ , según (4.40).

En las Figuras 5.25, 5.26 y 5.27 se representa la distribución de `perc_b_in_a` para una única pose de cámara y los seis métodos de registro considerados. Las Figuras 5.28, 5.29 y 5.30 muestran los resultados cuando la nube generada se obtiene a partir de la fusión de dos poses.

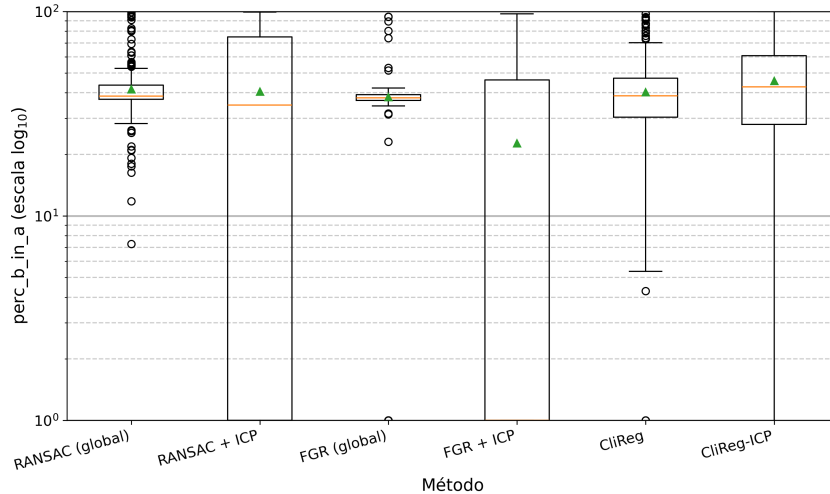


Figura 5.25 Distribución de `perc_b_in_a` para el registro del nivel bajo de la pieza con una única pose de cámara.

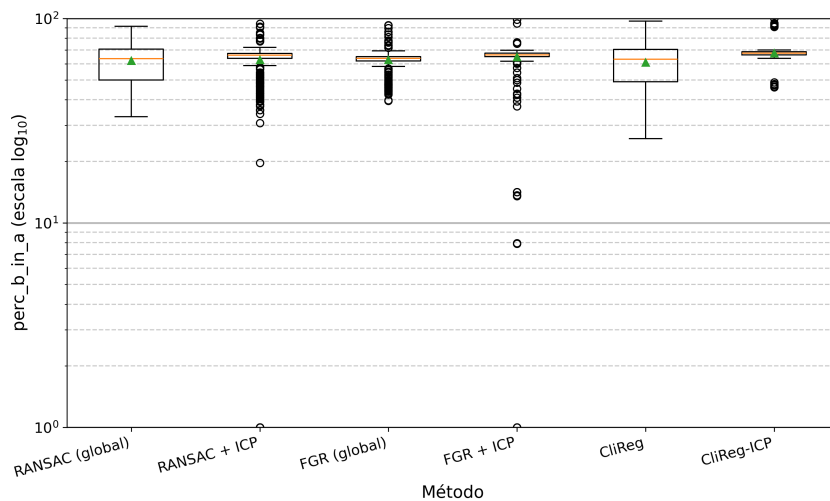


Figura 5.26 Distribución de `perc_b_in_a` para el registro del nivel bajo+medio de la pieza con una única pose de cámara.

En conjunto, las figuras muestran que `perc_b_in_a` es una métrica sensible; cualquier conjunto de puntos que quede fuera del volumen teórico, o un desplazamiento global de la nube generada, se traduce en porcentajes de coincidencia bajos. Al mismo tiempo, la métrica se beneficia claramente de la fusión de vistas, especialmente en niveles de impresión intermedios, donde la segunda pose permite completar regiones que en la captura simple permanecen ocultas.

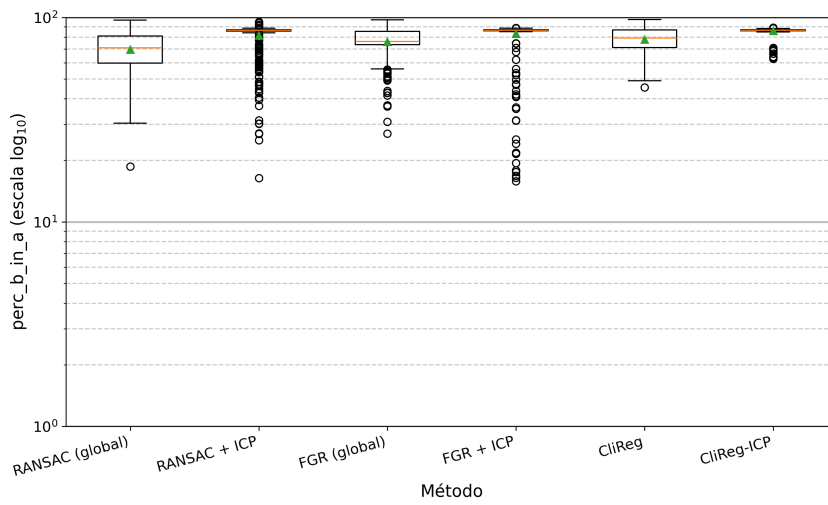


Figura 5.27 Distribución de `perc_b_in_a` para el registro de la pieza completa con una única pose de cámara.

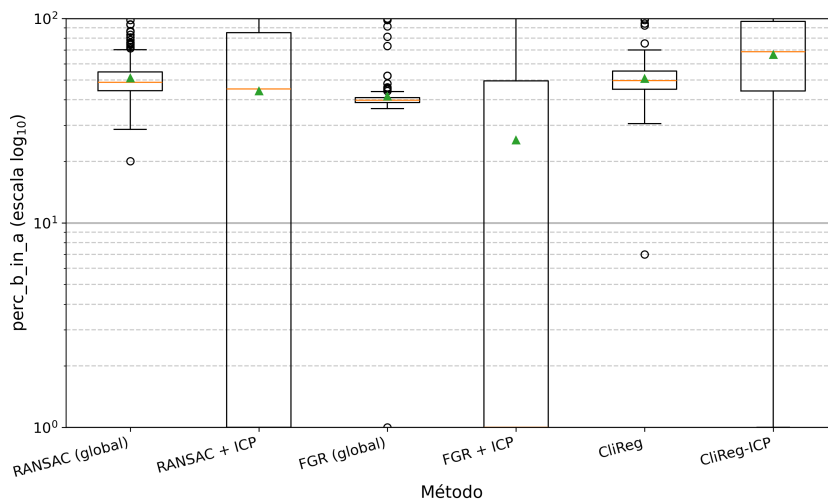


Figura 5.28 Distribución de `perc_b_in_a` para el registro del nivel bajo a partir de la fusión de dos poses de cámara.

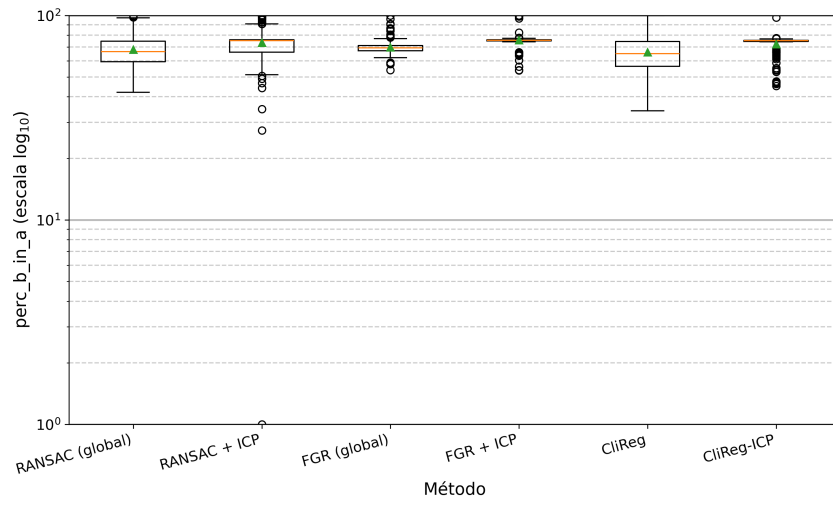


Figura 5.29 Distribución de `perc_b_in_a` para el registro del nivel bajo+medio a partir de la fusión de dos poses de cámara.

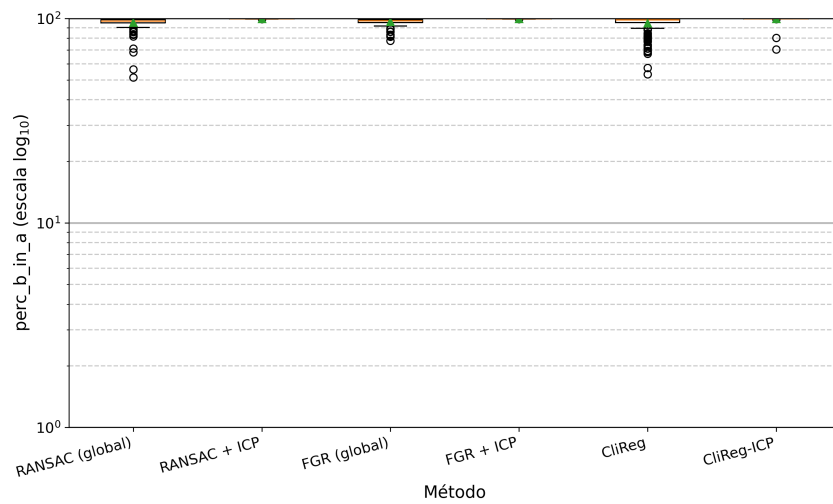


Figura 5.30 Distribución de `perc_b_in_a` para el registro de la pieza completa a partir de la fusión de dos poses de cámara.

Nivel 1: sección baja

En el nivel bajo de 19 % de altura, Figuras 5.25 y 5.28, las medianas con una única pose se sitúan en un rango intermedio, en torno al 30–40 %, con cajas relativamente anchas y presencia de valores atípicos tanto muy bajos como cercanos al 100 %. Este comportamiento refleja la sensibilidad de `perc_b.in_a` cuando sólo se ha impreso una fracción reducida de la pieza. Además, los métodos que combinan un registro global con un refinamiento ICP muestran una dispersión especialmente alta, lo que indica que los fallos de registro se traducen en porcentajes de coincidencia muy pequeños para algunas capturas.

Al introducir la fusión de dos poses, la distribución se desplaza hacia valores más altos para la mayoría de métodos. Las medianas aumentan moderadamente y, en especial para FGR y CliReg-ICP, la cola de valores muy bajos se reduce. La segunda vista permite capturar puntos del volumen teórico que quedaban ocultos con una única cámara, de modo que una fracción mayor de la nube generada se sitúa efectivamente dentro del modelo. No obstante, en este nivel la métrica sigue penalizando con fuerza cualquier grupo de puntos mal recortados o mal alineados, por lo que persisten outliers asociados a nubes con artefactos o registros fallidos.

Nivel 2: sección baja+media

Cuando la pieza alcanza el 52 % de altura, Figuras 5.26 y 5.29, las distribuciones con una sola pose se desplazan claramente hacia valores altos y las cajas se estrechan. La mayoría de métodos presentan medianas alrededor del 60 %, con menos casos extremos. La mayor cantidad de geometría impresa hace que el volumen observado por la cámara se parezca más al volumen teórico, de manera que la métrica empieza a ser más estable: la mayoría de adquisiciones muestran coincidencias volumétricas altas y sólo algunas capturas aisladas caen a valores muy bajos por errores de registro.

Con la fusión de dos poses, `perc_b.in_a` se obtienen medianas que rondan el 70 % y una variabilidad muy reducida.

Nivel 3: pieza completa

En el nivel completo, Figuras 5.27 y 5.30, `perc_b.in_a` se concentra en torno a valores muy próximos al 92 % para la mayoría de métodos, incluso con una única pose. Las cajas son estrechas y aparecen ciertos outliers asociados a situaciones de registro ligeramente desplazado o a la presencia de nubes fuera del volumen de la pieza. En este escenario la nube generada ocupa prácticamente el mismo volumen que el modelo, por lo que casi todos los puntos encuentran una pareja dentro de la tolerancia τ .

La fusión de dos poses incrementa razonablemente la mediana situandola en valores próximos del 100 %, y contribuye a reducir aún más la dispersión y a eliminar casos extremos. `perc_b_in_a` se comporta así como un indicador robusto de final de impresión: cuando la pieza está completa y el registro es correcto, la métrica se sitúa sistemáticamente cerca del 100 %, independientemente del método utilizado.

5.3.4. Coherencia entre métricas y lectura conjunta

Las tres métricas consideradas proporcionan perspectivas complementarias sobre el estado de la impresión. Mientras que `zmax_percent` resume el avance en términos de altura máxima, `progress_surface` evalúa la fracción de superficie teórica efectivamente observada y `perc_b_in_a` mide el grado en que el volumen ocupado por la nube generada queda contenido en el modelo. Analizar conjuntamente su comportamiento permite discriminar entre situaciones de impresión correcta y casos en los que el registro o la adquisición introducen errores significativos.

En los escenarios favorables, especialmente en el nivel 3 y en los casos fusionados de los niveles intermedios, las tres métricas tienden a coincidir: `zmax_percent` se sitúa cerca del porcentaje teórico de altura igual que `progress_surface` y `perc_b_in_a`. Esta concordancia indica que la nube registrada reproduce de forma coherente tanto la altura como la superficie y el volumen de la pieza, y refuerza la confianza en que la impresión ha alcanzado el estado esperado. En estos casos, pequeñas variaciones entre métodos de registro apenas afectan a la lectura global del proceso.

Las diferencias aparecen con mayor frecuencia en el nivel bajo y en aquellas capturas donde el registro falla parcialmente. Un primer patrón se observa cuando `zmax_percent` tiende a tomar valores cercanos o incluso superiores a la altura teórica, mientras que `progress_surface` y `perc_b_in_a` permanecen en rangos bajos. Esta situación suele deberse a ruido de la cámara o a puntos aislados por encima de la superficie real que disparan la altura máxima, sin que exista una coincidencia real de la geometría con el modelo.

Un segundo patrón se produce cuando `progress_surface` y `perc_b_in_a` divergen entre sí pese a que `zmax_percent` sea coherente con la altura teórica.

En la mayoría de escenarios fusionados, `zmax_percent` apenas varía, ya que la altura máxima viene determinada por el propio estado de la pieza, mientras que por otro lado `progress_surface` y `perc_b_in_a` mejoran de forma clara al disponer de más puntos y menos oclusiones. Cuando esta mejora no se produce, o incluso se degrada alguna de las métricas geométricas, suele ser indicio de problemas de registro entre vistas o de acumulación de ruido en la nube fusionada.

A partir de estos patrones, se propone un uso combinado de las métricas en el proceso

de validación. La métrica `zmax_percent` puede emplearse como un indicador del estado de avance. Es sencilla de calcular, poco sensible a los detalles de filtrado y permite seguir en tiempo real la progresión global de la altura impresa. Las métricas `progress_surface` y `perc_b_in_ase` reservan para una validación geométrica detallada de impresiones que presentan un avance entre medio y completo. Especialmente se plantea el uso en conjunto para capturas de múltiples poses. De este modo, la combinación de un indicador rápido de avance con métricas de superficie y volumen permite realizar un seguimiento robusto del proceso de impresión.

Conclusión y Trabajo Futuro

Este capítulo cierra el trabajo resumiendo los principales aportes del sistema propuesto para la validación geométrica del proceso de impresión 3D, y discutiendo su desempeño según los resultados experimentales obtenidos. En primer lugar, se resume el desarrollo realizado y se evalúa el grado de cumplimiento de los objetivos planteados, identificando los elementos del sistema que resultaron determinantes para la estabilidad de la estimación. A continuación, se exponen las limitaciones observadas en adquisición, normalización, registro y métricas, delimitando el alcance actual del sistema. Finalmente, se proponen líneas de trabajo futuro orientadas a incrementar la robustez y la automatización, con el fin de acercar la herramienta a un escenario de monitoreo en línea aplicable a condiciones reales de fabricación.

6.1. Síntesis del trabajo realizado

En este trabajo se aborda el problema de la validación del proceso de impresión 3D a partir de información geométrica adquirida durante la fabricación. Para esto, se propone un sistema capaz de estimar el estado de avance de una pieza impresa comparando la nube de puntos observada y un modelo teórico de referencia, con el objetivo de disponer de un criterio cuantitativo y reproducible sobre la evolución de la impresión.

Para ello, se ha desarrollado un sistema basado en cuatro etapas principales:

- Adquisición RGB-D de la escena mediante un sensor de profundidad.
- Generación y preprocesado de las nubes de puntos para mejorar la calidad geométrica y la consistencia entre capturas.
- Registro de la nube observada con respecto al modelo teórico.

- Definición de métricas de progreso que permitan interpretar el estado de impresión de forma clara.

Todo el sistema se ha planteado con una arquitectura modular, de manera que cada bloque resuelve una función bien definida y su integración permite ejecutar el flujo completo de forma coherente, desde la captura hasta la estimación final del progreso. Esta separación por módulos facilita tanto la depuración del sistema como su extensión con nuevas estrategias de filtrado, registro o métricas, sin alterar el resto del flujo.

Además, se ha incorporado una interfaz gráfica de usuario que encapsula los pasos principales del proceso y permite ejecutar el sistema sin depender de la línea de comandos. De este modo, el trabajo no sólo culmina en un conjunto de algoritmos, sino en una herramienta operativa que integra adquisición, alineamiento, cálculo de métricas y generación de salidas visuales y numéricas.

6.2. Cumplimiento de los objetivos planteados

El objetivo general de este trabajo consistió en diseñar e implementar un sistema de adquisición y análisis de nubes de puntos que permitiera comparar la geometría observada durante la impresión con la geometría teórica de referencia, con el fin de estimar el progreso y evaluar la coherencia del proceso. A partir de este objetivo global, se definieron objetivos específicos agrupados en adquisición, registro, estimación del progreso y validación experimental.

Adquisición de datos: El módulo de adquisición se desarrolló con la funcionalidad de capturar información de color y profundidad de manera consistente, generando nubes de puntos representativas de la escena. También se estableció un procedimiento para estructurar las capturas y almacenar la información asociada necesaria para las etapas posteriores. En conjunto, esta parte cumple el objetivo de proporcionar datos tridimensionales adecuados para el análisis, tal y como se evidencia en los resultados obtenidos.

Preprocesamiento de nubes de puntos: Se desarrolló una etapa de preprocesamiento orientada a limpiar las nubes adquiridas. Dentro de este módulo se reduce el ruido y se eliminan puntos que no aportan información. Este bloque permite obtener nubes comparables con el modelo teórico, y representa un paso clave para mejorar la coherencia del proceso de alineación.

Registro de nubes de puntos: Se implementó un flujo que permite llevar la nube observada a un marco comparable con el modelo teórico usando diferentes algoritmos y combinaciones. Los resultados muestran que el registro es un paso determinante para la estabilidad del sistema, y que el enfoque desarrollado permite obtener alineaciones

coherentes en los escenarios planteados, de acuerdo con el análisis presentado en el capítulo de resultados y discusión.

Métricas de progreso: Se definieron e integraron métricas orientadas a estimar el estado de avance de la impresión desde distintas perspectivas geométricas, de forma que la estimación no dependa de un único indicador. En los experimentos se observa que estas métricas capturan tendencias y aportan interpretabilidad al sistema, lo que permite utilizarlas como base para una validación cuantitativa de la impresión.

Validación experimental: Finalmente, se diseñó una metodología de evaluación con distintos niveles de progresión y configuraciones de captura, con el objetivo de someter el sistema a condiciones representativas del proceso de impresión. En particular, se definieron tres estados de fabricación, sección baja, sección baja+media y pieza completa, y para cada uno se adquirieron capturas desde dos poses de cámara complementarias, además de un escenario adicional basado en la fusión de vistas. Esta estrategia permitió analizar el impacto de la cantidad de geometría disponible y la sensibilidad de las métricas frente a oclusiones, ruido y variaciones entre capturas.

Los resultados experimentales muestran que la robustez del sistema depende del grado de progresión de la pieza. En escenarios tempranos, con poca altura impresa, aparecen fallos de registro y alta variabilidad, especialmente al aplicar refinamientos locales con ICP. Esto limita la fiabilidad de las métricas en esa fase. Sin embargo, a partir de un nivel intermedio de fabricación, el registro converge de forma estable en la mayoría de capturas y las métricas se vuelven consistentes: `zmax_percent` refleja correctamente el avance en altura, mientras que `progress_surface` y `perc_b_in_a` aportan una validación más exigente basada en cobertura superficial y coincidencia volumétrica. Adicionalmente, la fusión de dos poses tiende a mejorar las métricas geométricas al reducir oclusiones y aumentar la fracción de modelo observable, aunque también puede introducir ruido acumulado que debe ser gestionado por el preprocesado.

En conjunto, la metodología y los experimentos realizados permiten afirmar que los objetivos de validación experimental se han cumplido, proporcionando un análisis reproducible del comportamiento del sistema bajo distintos niveles de progresión y configuraciones de captura.

6.3. Limitaciones del sistema

A pesar de los resultados obtenidos, el sistema presenta limitaciones tanto en el proceso de adquisición como a la naturaleza del registro geométrico:

- **Dependencia de la geometría disponible:** en etapas tempranas de impresión,

la superficie observada es reducida, lo que incrementa la variabilidad del registro y de las métricas.

- **Sensibilidad al ruido del sensor RGB-D:** la presencia de puntos flotantes, discontinuidades de profundidad y artefactos en bordes puede sesgar métricas basadas en máximos, como `zmax_percent`, y degradar métricas de superficie y volumen si el filtrado no es suficiente.
- **Dependencia de la referencia por ArUco:** la normalización espacial requiere visibilidad suficiente de marcadores y estabilidad en su detección. Oclusiones o detecciones parciales pueden afectar la consistencia del marco alineado, que por extensión afecta al registro.
- **Fusión multivista:** aunque mejora la completitud geométrica, puede incrementar el error acumulado por pequeñas discrepancias entre poses, solapes mal alineados o duplicidades, requiriendo estrategias de depuración y filtrado más estrictas.

Estas limitaciones no invalidan el enfoque propuesto, pero delimitan la fiabilidad del sistema. Además, justifican la necesidad de una lectura conjunta de métricas.

6.4. Líneas de trabajo futuras

Como continuación de este trabajo, se proponen varias líneas de mejora orientadas a aumentar la robustez, automatización y aplicabilidad del sistema:

- **Seguimiento temporal del progreso:** incorporar un filtrado temporal o un esquema de estimación recursivo que integre información entre instantes consecutivos, reduciendo la sensibilidad a capturas aisladas y estabilizando métricas a lo largo del tiempo.
- **Mejoras en fusión multivista:** integrar estrategias de fusión con control de duplicidades, consistencia de solapes y depuración basada en calidad de profundidad, para explotar múltiples poses sin penalizar por acumulación de ruido.
- **Criterios automáticos de aceptación/rechazo:** definir umbrales o reglas de decisión basadas en métricas de convergencia para descartar registros fallidos antes de interpretar `zmax_percent`, `progress_surface` o `perc_b_in_a`.
- **Ampliación del conjunto de piezas y condiciones:** extender la validación a geometrías más complejas, variaciones de material y condiciones de iluminación, para caracterizar mejor el comportamiento del sistema en escenarios reales.

6.5. Conclusión final

El trabajo desarrollado demuestra la viabilidad de validar el avance de una impresión 3D mediante la comparación geométrica entre una nube de puntos adquirida con un sensor RGB-D y un modelo teórico de referencia. La arquitectura modular implementada permite ejecutar de forma coherente las etapas de adquisición, preprocesado, registro, evaluación y visualización, proporcionando métricas interpretables que capturan el progreso desde perspectivas complementarias.

Los resultados experimentales indican que, cuando la impresión alcanza un grado intermedio o alto de progresión, el sistema converge de forma estable y ofrece estimaciones fiables del estado de la pieza. Especialmente cuando se dispone de múltiples vistas. De este modo, el sistema propuesto sienta una base para futuras extensiones orientadas a una monitorización más robusta y automatizada del proceso de impresión.

Bibliografía

- [1] O. Abdulhameed, A. Al-Ahmari, W. Ameen y S. H. Mian, “Additive manufacturing: Challenges, trends, and applications”, *Advances in Mechanical Engineering*, vol. 11, n.º 2, feb. de 2019, ISSN: 16878140. DOI: 10.1177/1687814018822880.
- [2] T. Ma, Y. Zhang, K. Ruan, H. Guo, M. He, X. Shi, Y. Guo, J. Kong y J. Gu, “Advances in 3D printing for polymer composites: A review”, *InfoMat*, vol. 6, n.º 6, jun. de 2024, ISSN: 25673165. DOI: 10.1002/inf2.12568.
- [3] S. Wickramasinghe, T. Do y P. Tran, “FDM-Based 3D printing of polymer and associated composite: A review on mechanical properties, defects and treatments”, *Polymers*, vol. 12, n.º 7, págs. 1-42, jul. de 2020, ISSN: 20734360. DOI: 10.3390/polym12071529. visitado 23 de sep. de 2025. dirección: <https://www.mdpi.com/2073-4360/12/7/1529>.
- [4] Y. Tian, C. X. Chen, X. Xu, J. Wang, X. Hou, K. Li, X. Lu, H. Y. Shi, E. S. Lee y H. B. Jiang, “A Review of 3D Printing in Dentistry: Technologies, Affecting Factors, and Applications”, *Scanning*, vol. 2021, 2021, ISSN: 19328745. DOI: 10.1155/2021/9950131.
- [5] Repsol, *Impresión 3D*, Publication Title: Repsol, 2025. visitado 23 de sep. de 2025. dirección: <https://www.repsol.com/es/energia-avanzar/innovacion/impresion-3d/index.cshtml>.
- [6] Autodesk Inc., *Impresión 3D soluciones de Autodesk*, Publication Title: Autodesk, 2025. visitado 23 de sep. de 2025. dirección: <https://www.autodesk.com/es/solutions/3d-printing>.
- [7] G. D. Goh, S. L. Sing y W. Y. Yeong, “A review on machine learning in 3D printing: applications, potential, and challenges”, *Artificial Intelligence Review*, vol. 54, n.º 1, págs. 63-94, ene. de 2021, ISSN: 15737462. DOI: 10.1007/s10462-020-09876-9.

- [8] F. Blais, “Review of 20 years of range sensor development”, en, *Journal of Electronic Imaging*, vol. 13, n.º 1, pág. 231, ene. de 2004, ISSN: 1017-9909. DOI: 10.1117/1.1631921. visitado 18 de ene. de 2026. dirección: <http://electronicimaging.spiedigitallibrary.org/article.aspx?doi=10.1117/1.1631921>.
- [9] M. Westoby, J. Brasington, N. Glasser, M. Hambrey y J. Reynolds, “‘Structure-from-Motion’ photogrammetry: A low-cost, effective tool for geoscience applications”, en, *Geomorphology*, vol. 179, págs. 300-314, dic. de 2012, ISSN: 0169555X. DOI: 10.1016/j.geomorph.2012.08.021. visitado 18 de ene. de 2026. dirección: <https://linkinghub.elsevier.com/retrieve/pii/S0169555X12004217>.
- [10] K. Khoshelham y S. O. Elberink, “Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications”, en, *Sensors*, vol. 12, n.º 2, págs. 1437-1454, feb. de 2012, ISSN: 1424-8220. DOI: 10.3390/s120201437. visitado 18 de ene. de 2026. dirección: <https://www.mdpi.com/1424-8220/12/2/1437>.
- [11] S. Foix, G. Alenya y C. Torras, “Lock-in Time-of-Flight (ToF) Cameras: A Survey”, *IEEE Sensors Journal*, vol. 11, n.º 9, págs. 1917-1926, sep. de 2011, ISSN: 1530-437X, 1558-1748. DOI: 10.1109/JSEN.2010.2101060. visitado 18 de ene. de 2026. dirección: <http://ieeexplore.ieee.org/document/5686908/>.
- [12] H. Sarbolandi, D. Lefloch y A. Kolb, “Kinect range sensing: Structured-light versus Time-of-Flight Kinect”, en, *Computer Vision and Image Understanding*, vol. 139, págs. 1-20, oct. de 2015, ISSN: 10773142. DOI: 10.1016/j.cviu.2015.05.006. visitado 18 de ene. de 2026. dirección: <https://linkinghub.elsevier.com/retrieve/pii/S1077314215001071>.
- [13] Z. Zhang, “A flexible new technique for camera calibration”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, n.º 11, págs. 1330-1334, 2000, ISSN: 01628828. DOI: 10.1109/34.888718.
- [14] F. Basso, E. Menegatti y A. Pretto, “Robust Intrinsic and Extrinsic Calibration of RGB-D Cameras”, *IEEE Transactions on Robotics*, vol. 34, n.º 5, págs. 1315-1332, oct. de 2018, ISSN: 1552-3098. DOI: 10.1109/TR0.2018.2853742.
- [15] H. Afzal, D. Aouada, D. Font, B. Mirbach y B. Ottersten, “RGB-D Multi-view System Calibration for Full 3D Scene Reconstruction”, en *2014 22nd International Conference on Pattern Recognition*, IEEE, ago. de 2014, págs. 2459-2464, ISBN: 978-1-4799-5209-0. DOI: 10.1109/ICPR.2014.425.
- [16] F. Endres, J. Hess, J. Sturm, D. Cremers y W. Burgard, “3-D Mapping With an RGB-D Camera”, *IEEE Transactions on Robotics*, vol. 30, n.º 1, págs. 177-187, feb. de 2014, ISSN: 1552-3098. DOI: 10.1109/TR0.2013.2279412.

- [17] R. B. Rusu y S. Cousins, “3D is here: Point Cloud Library (PCL)”, en *2011 IEEE International Conference on Robotics and Automation*, IEEE, mayo de 2011, págs. 1-4, ISBN: 978-1-61284-386-5. DOI: 10.1109/ICRA.2011.5980567.
- [18] A. Zabatani, V. Surazhsky, E. Sperling, S. B. Moshe, O. Menashe, D. H. Silver, Z. Karni, A. M. Bronstein, M. M. Bronstein y R. Kimmel, “Intel® RealSense™ SR300 Coded Light Depth Camera”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, n.º 10, págs. 2333-2345, oct. de 2020, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2019.2915841.
- [19] M. Servi, A. Profili, R. Furferi e Y. Volpe, “Comparative Evaluation of Intel RealSense D415, D435i, D455, and Microsoft Azure Kinect DK Sensors for 3D Vision Applications”, *IEEE Access*, vol. 12, págs. 111 311-111 321, 2024, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2024.3441238.
- [20] R. A. Newcombe, A. Fitzgibbon, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton y S. Hodges, “KinectFusion: Real-time dense surface mapping and tracking”, en *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, IEEE, oct. de 2011, págs. 127-136, ISBN: 978-1-4577-2183-0. DOI: 10.1109/ISMAR.2011.6092378.
- [21] S. Song, S. P. Lichtenberg y J. Xiao, “SUN RGB-D: A RGB-D scene understanding benchmark suite”, en *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, jun. de 2015, págs. 567-576, ISBN: 978-1-4673-6964-0. DOI: 10.1109/CVPR.2015.7298655.
- [22] E. Olson, “AprilTag: A robust and flexible visual fiducial system”, en *2011 IEEE International Conference on Robotics and Automation*, IEEE, mayo de 2011, págs. 3400-3407, ISBN: 978-1-61284-386-5. DOI: 10.1109/ICRA.2011.5979561.
- [23] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton y A. Frenkel, “On the segmentation of 3D LIDAR point clouds”, en *2011 IEEE International Conference on Robotics and Automation*, IEEE, mayo de 2011, págs. 2798-2805, ISBN: 978-1-61284-386-5. DOI: 10.1109/ICRA.2011.5979818.
- [24] R. B. Rusu, N. Blodow y M. Beetz, “Fast Point Feature Histograms (FPFH) for 3D registration”, en *2009 IEEE International Conference on Robotics and Automation*, IEEE, mayo de 2009, págs. 3212-3217, ISBN: 978-1-4244-2788-8. DOI: 10.1109/ROBOT.2009.5152473.
- [25] J. Prankl, A. Aldoma, A. Svejda y M. Vincze, “RGB-D object modelling for object recognition and tracking”, en *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, sep. de 2015, págs. 96-103, ISBN: 978-1-4799-9994-1. DOI: 10.1109/IROS.2015.7353360.

- [26] S. Rusinkiewicz y M. Levoy, “Efficient variants of the ICP algorithm”, en *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, IEEE Comput. Soc, págs. 145-152, ISBN: 0-7695-0984-3. DOI: 10.1109/IM.2001.924423.
- [27] P. Besl y N. D. McKay, “A method for registration of 3-D shapes”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, n.º 2, págs. 239-256, feb. de 1992, ISSN: 0162-8828, 2160-9292. DOI: 10.1109/34.121791. visitado 9 de dic. de 2025. dirección: <http://ieeexplore.ieee.org/document/121791/>.
- [28] H. Yang, J. Shi y L. Carlone, “TEASER: Fast and Certifiable Point Cloud Registration”, *IEEE Transactions on Robotics*, vol. 37, n.º 2, págs. 314-333, abr. de 2021, ISSN: 1552-3098. DOI: 10.1109/TR0.2020.3033695.
- [29] J. Yang, H. Li, D. Campbell e Y. Jia, “Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, n.º 11, págs. 2241-2254, nov. de 2016, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2015.2513405.
- [30] J. Laserna, P. S. Segundo y D. Álvarez, “CliReg: Clique-Based Robust Point Cloud Registration”, *IEEE Transactions on Robotics*, vol. 41, págs. 1898-1917, 2025, ISSN: 1552-3098. DOI: 10.1109/TR0.2025.3542954.
- [31] Z. Ye, C. Liu, W. Tian y C. Kan, “In-situ point cloud fusion for layer-wise monitoring of additive manufacturing”, en *Journal of Manufacturing Systems*, vol. 61, págs. 210-222, oct. de 2021, ISSN: 02786125. DOI: 10.1016/j.jmsy.2021.09.002. visitado 6 de oct. de 2025. dirección: <https://linkinghub.elsevier.com/retrieve/pii/S0278612521001886>.
- [32] Z. Ye, C. Liu, W. Tian y C. Kan, “A Deep Learning Approach for the Identification of Small Process Shifts in Additive Manufacturing using 3D Point Clouds”, en *Procedia Manufacturing*, vol. 48, págs. 770-775, 2020, ISSN: 23519789. DOI: 10.1016/j.promfg.2020.05.112. visitado 6 de oct. de 2025. dirección: <https://linkinghub.elsevier.com/retrieve/pii/S235197892031564X>.
- [33] H. Yu, C. Peng, Z. Zhao, L. Bai y J. Han, “Visual Texture-Based 3-D Roughness Measurement for Additive Manufacturing Surfaces”, *IEEE Access*, vol. 7, págs. 186 646-186 656, 2019, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2919682. visitado 6 de oct. de 2025. dirección: <https://ieeexplore.ieee.org/document/8725498/>.
- [34] X. Zhang, Y. Zheng, V. Suresh, S. Wang, Q. Li, B. Li y H. Qin, “Correlation approach for quality assurance of additive manufactured parts based on optical metrology”, en *Journal of Manufacturing Processes*, vol. 53, págs. 310-317, mayo de 2020, ISSN: 15266125. DOI: 10.1016/j.jmapro.2020.02.037. visitado 6 de oct. de 2025. dirección: <https://linkinghub.elsevier.com/retrieve/pii/S1526612520301171>.

- [35] C. Lequn, Y. Xiling, X. Peng, K. M. Seung y B. Guijun, “Surface Monitoring for Additive Manufacturing with in-situ Point Cloud Processing”, 2020, ISBN: 978-1-7281-6139-6. DOI: 10.1109/ICCAR49639.2020.9108092.
- [36] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas y M. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion”, en *Pattern Recognition*, vol. 47, n.º 6, págs. 2280-2292, jun. de 2014, ISSN: 00313203. DOI: 10.1016/j.patcog.2014.01.005. visitado 9 de dic. de 2025. dirección: <https://linkinghub.elsevier.com/retrieve/pii/S0031320314000235>.
- [37] R. B. Rusu, “Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments”, en *KI - Künstliche Intelligenz*, vol. 24, n.º 4, págs. 345-348, nov. de 2010, ISSN: 0933-1875, 1610-1987. DOI: 10.1007/s13218-010-0059-6. visitado 9 de dic. de 2025. dirección: <http://link.springer.com/10.1007/s13218-010-0059-6>.
- [38] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald y W. Stuetzle, “Surface reconstruction from unorganized points”, en *ACM SIGGRAPH Computer Graphics*, vol. 26, n.º 2, págs. 71-78, jul. de 1992, ISSN: 0097-8930. DOI: 10.1145/142920.134011. visitado 9 de dic. de 2025. dirección: <https://dl.acm.org/doi/10.1145/142920.134011>.
- [39] C. Tomasi y R. Manduchi, “Bilateral filtering for gray and color images”, en *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, Bombay, India: Narosa Publishing House, 1998, págs. 839-846, ISBN: 978-81-7319-221-0. DOI: 10.1109/ICCV.1998.710815. visitado 9 de dic. de 2025. dirección: <http://ieeexplore.ieee.org/document/710815/>.
- [40] S. Fleishman, I. Drori y D. Cohen-Or, “Bilateral mesh denoising”, en *ACM Transactions on Graphics*, vol. 22, n.º 3, págs. 950-953, jul. de 2003, ISSN: 0730-0301, 1557-7368. DOI: 10.1145/882262.882368. visitado 9 de dic. de 2025. dirección: <https://dl.acm.org/doi/10.1145/882262.882368>.
- [41] S. Katz, A. Tal y R. Basri, “Direct visibility of point sets”, en *ACM Transactions on Graphics*, vol. 26, n.º 3, pág. 24, jul. de 2007, ISSN: 0730-0301, 1557-7368. DOI: 10.1145/1276377.1276407. visitado 9 de dic. de 2025. dirección: <https://dl.acm.org/doi/10.1145/1276377.1276407>.
- [42] I. Vizzo, X. Chen, N. Chebrolu, J. Behley y C. Stachniss, “Poisson Surface Reconstruction for LiDAR Odometry and Mapping”, en *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi’an, China: IEEE, mayo de 2021, págs. 5624-5630, ISBN: 978-1-7281-9077-8. DOI: 10.1109/ICRA48506.2021.9562069. visitado 9 de dic. de 2025. dirección: <https://ieeexplore.ieee.org/document/9562069/>.

- [43] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin y D. Jacobs, “A search engine for 3D models”, en, *ACM Transactions on Graphics*, vol. 22, n.º 1, págs. 83-105, ene. de 2003, ISSN: 0730-0301, 1557-7368. DOI: 10.1145/588272.588279. visitado 9 de dic. de 2025. dirección: <https://dl.acm.org/doi/10.1145/588272.588279>.
- [44] “FAST APPROXIMATE NEAREST NEIGHBORS WITH AUTOMATIC ALGORITHM CONFIGURATION:” en *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications*, Lisboa, Portugal: SciTePress - Science, 2009, págs. 331-340, ISBN: 978-989-8111-69-2. DOI: 10.5220/0001787803310340. visitado 9 de dic. de 2025. dirección: <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0001787803310340>.

**Diagrama UML de los módulos del
programa**

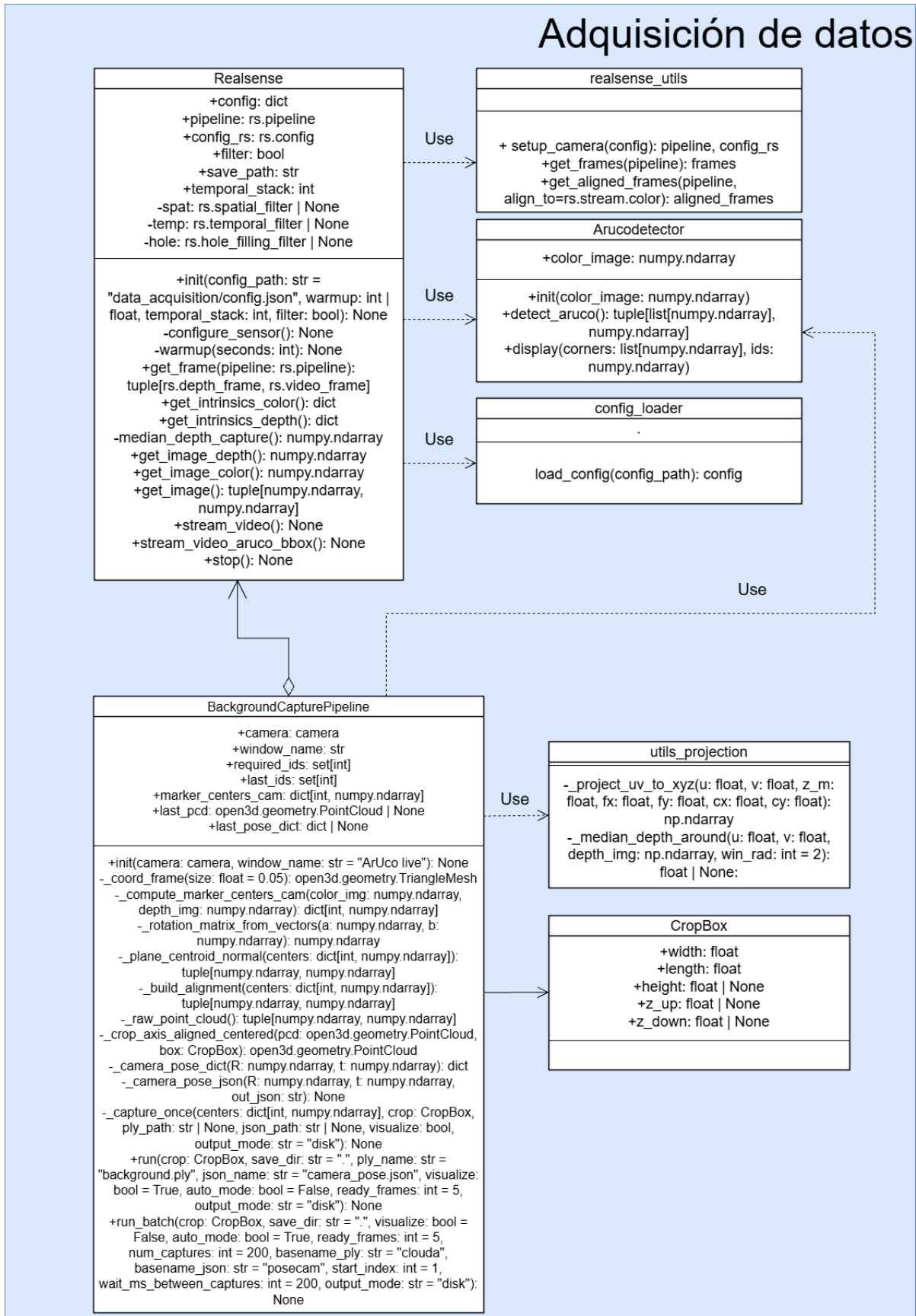


Figura A.1 Módulo de adquisición de datos

Representación geométrica

PointCloudProjector
+pcd: open3d.geometry.PointCloud -_pts: numpy.ndarray -_PLANE_AXES: dict[Literal["XY", "XZ", "YZ"], tuple[int, int]] -_PLANE_LABELS: dict[Literal["XY", "XZ", "YZ"], tuple[str, str]]
+init(pcd: open3d.geometry.PointCloud): None +points(): numpy.ndarray +project_to_plane(plane: Literal["XY", "XZ", "YZ"]): tuple[numpy.ndarray, tuple[str, str]] +compute_bounds(proj: numpy.ndarray, padding: float = 0.0): tuple[tuple[float, float], tuple[float, float]] +to_image(proj: numpy.ndarray, resolution: tuple[int, int] = (512, 512), bounds: Optional[tuple[tuple[float, float], tuple[float, float]]] = None, weights: Optional[numpy.ndarray] = None, normalize: bool = True, return_edges: bool = False): tuple[numpy.ndarray, Optional[tuple[numpy.ndarray, numpy.ndarray]]] +project_and_image(plane: Literal["XY", "XZ", "YZ"], resolution: tuple[int, int] = (512, 512), padding: float = 0.02, weights: Optional[numpy.ndarray] = None, normalize: bool = True, return_labels: bool = False): tuple +show_all_projections(mode: str = "scatter", resolution: tuple[int, int] = (512, 512), weights: Optional[numpy.ndarray] = None, padding: float = 0.02, block: bool = True, title_prefix: Optional[str] = None): list[matplotlib.figure.Figure]

HybridPointCloud
+voxel_size: float +axis_size: float +mesh: open3d.geometry.TriangleMesh None +point_cloud: open3d.geometry.PointCloud None +voxel_cloud: open3d.geometry.PointCloud None +origin: str
+init(source: str pathlib.Path open3d.geometry.PointCloud, num_points: int, voxel_size: float, autoscale_mm_to_m: bool, axis_size: float): None -load_mesh(stl_path: pathlib.Path, autoscale_mm_to_m: bool): None -normalize_position(): None -sample_points(num_points: int): None -voxelize(): None +save_point_cloud(path: str pathlib.Path, use_voxel: bool = True): None +visualize(use_voxel: bool = True, extra_clouds: list[open3d.geometry.PointCloud] None = None): None +repr(): str

Figura A.2 Módulo de representación geométrica

Preprocesamiento de nubes

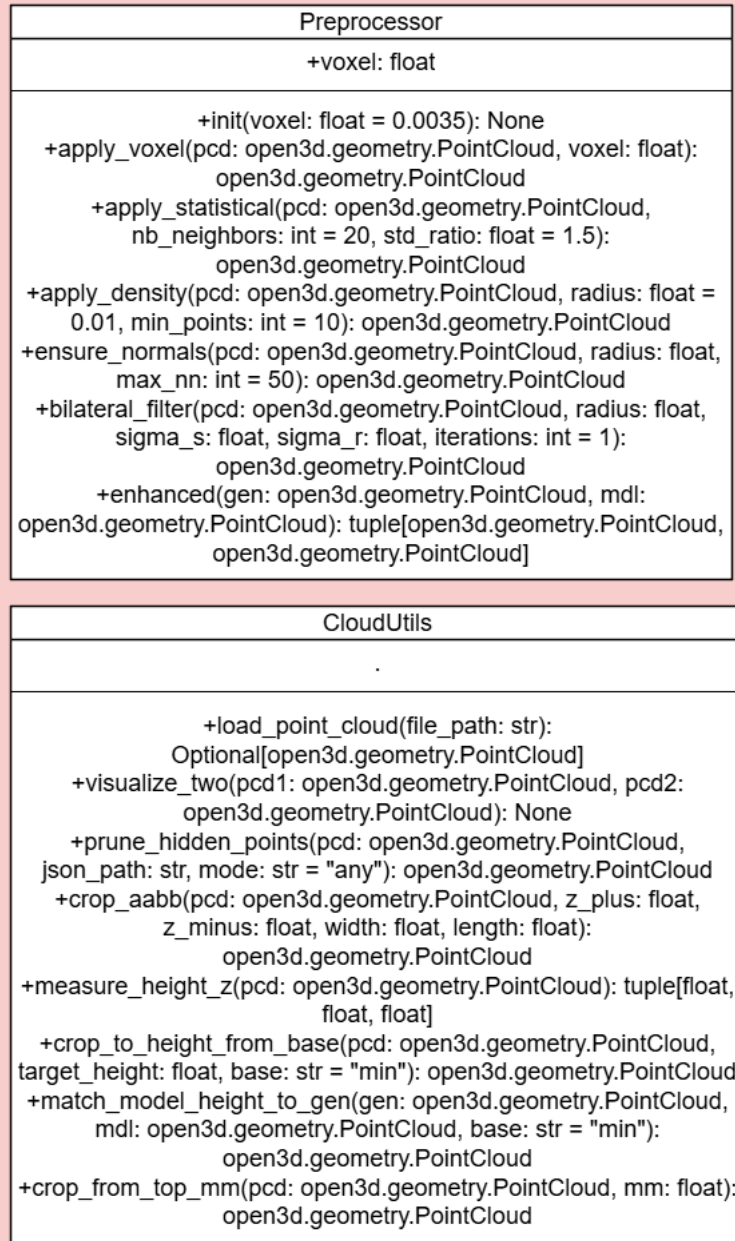


Figura A.3 Módulo de preprocesamiento de nubes

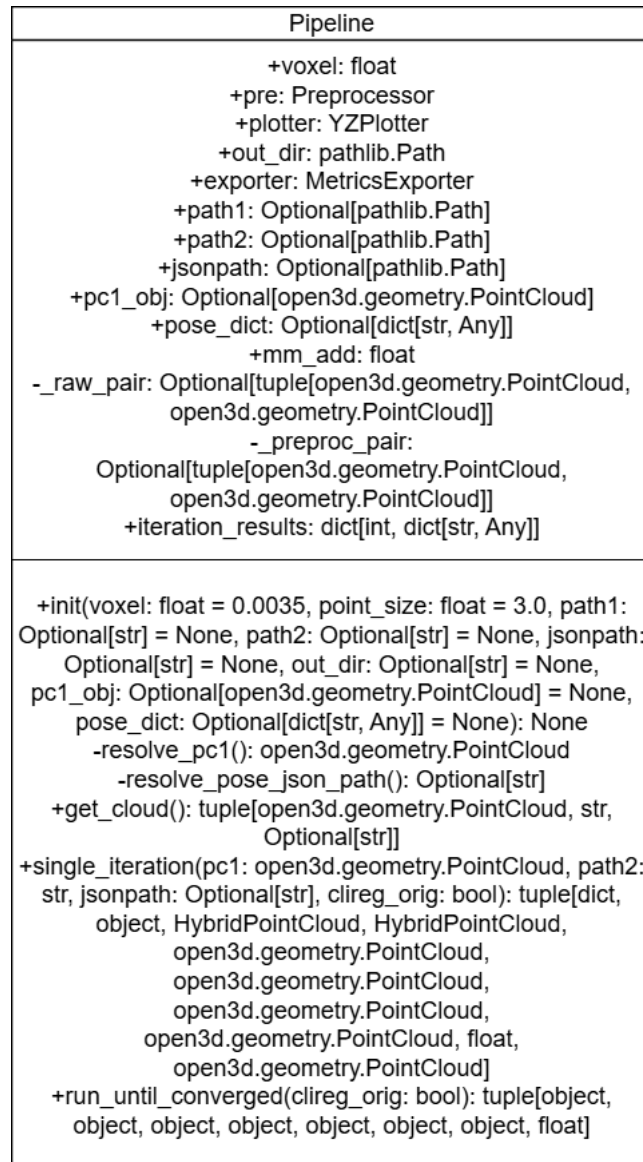


Figura A.4 Modulo del Pipeline

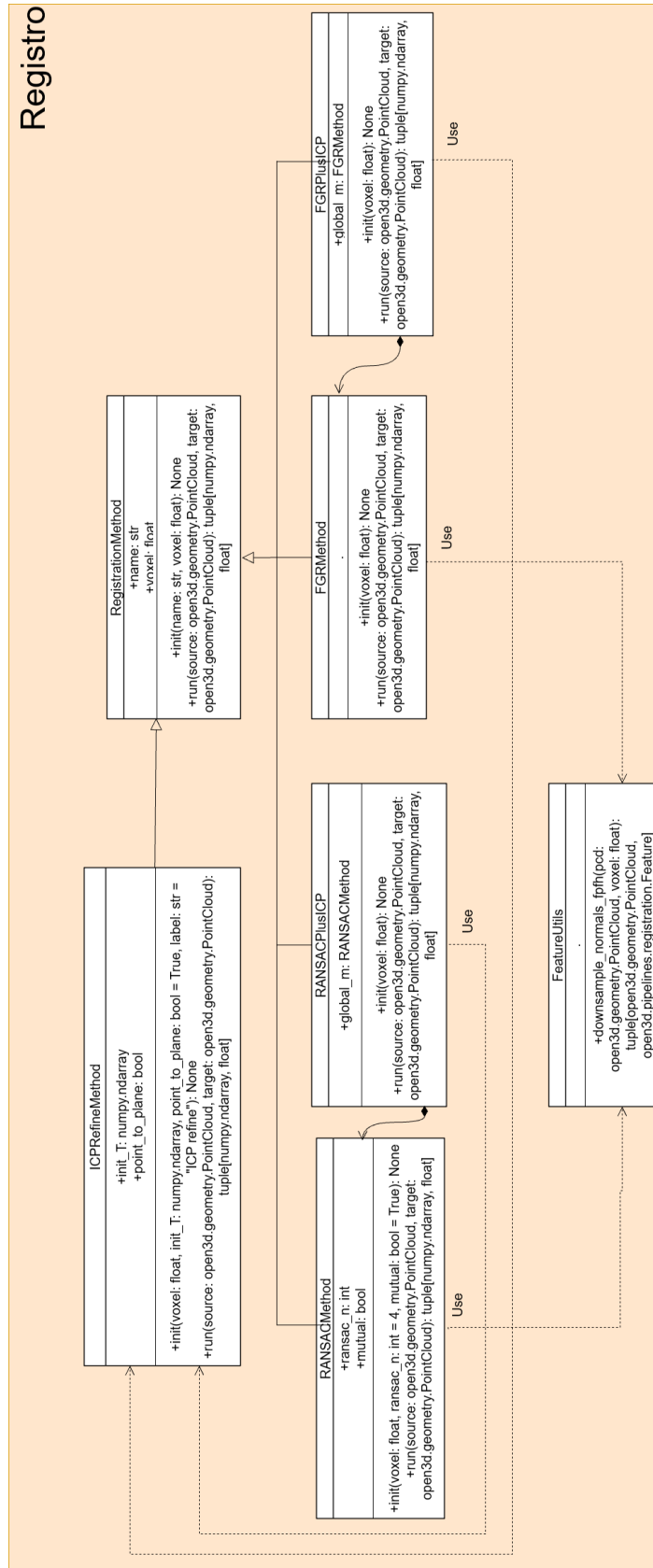


Figura A.5 Módulo de registro de nubes de puntos

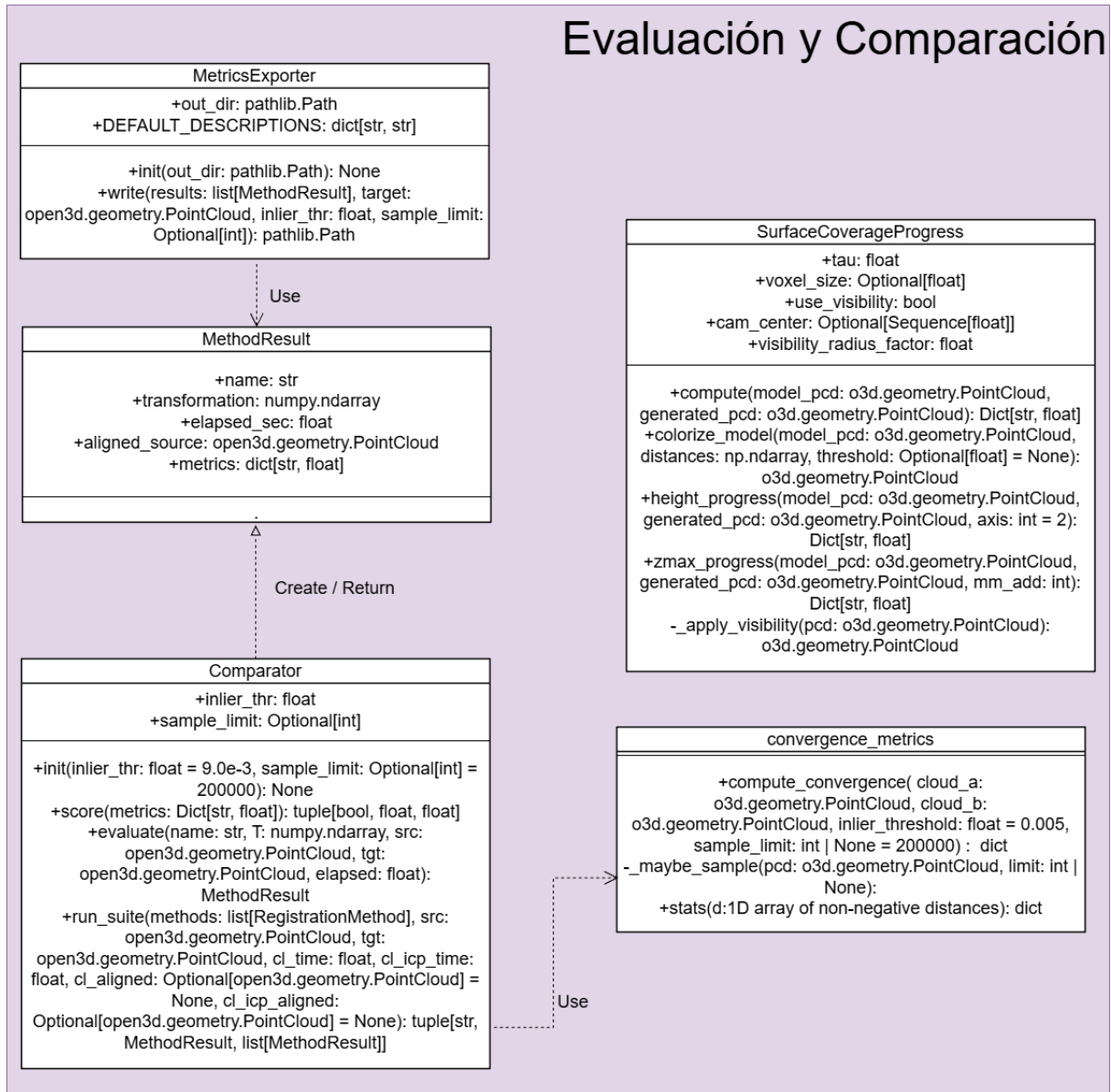


Figura A.6 Módulo de evaluación y comparación

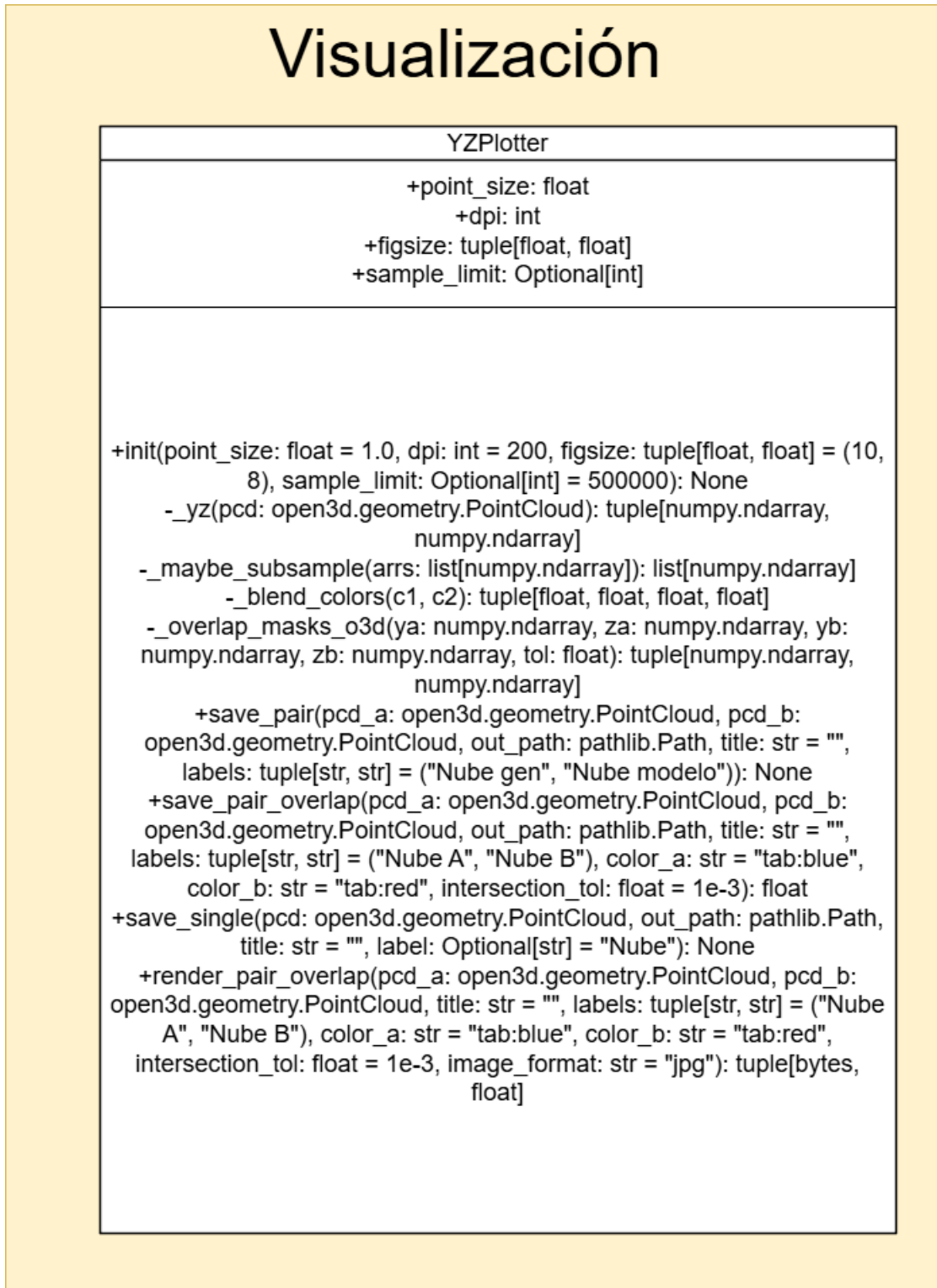


Figura A.7 Módulo de visualización

Análisis de aplicación e impacto

Este capítulo resume las aplicaciones del sistema desarrollado y evalúa sus impactos. Además, analiza los aspectos éticos pertinentes, su contribución a los Objetivos de Desarrollo Sostenible, el marco normativo de aplicación y los estándares que afectan al desarrollo, así como posibles contribuciones del trabajo a dichos estándares y buenas prácticas.

B.1. Aplicaciones

En este trabajo se ha desarrollado un sistema orientado a la monitorización geométrica de procesos de impresión 3D, principalmente FDM. En estos procesos es frecuente la aparición de defectos como falta de adhesión entre capas, rugosidad superficial y desviaciones dimensionales.

En muchos entornos de fabricación, la inspección continúa siendo principalmente reactiva al final del proceso, lo que incrementa el coste de corrección y puede implicar el rechazo completo de la pieza. En este contexto, el enfoque propuesto permite interpretar la pieza durante la impresión, comparando la geometría observada con el modelo teórico.

Una aplicación directa es la estimación cuantitativa del progreso basada en información 3D, alternativa a indicadores habituales como el tiempo transcurrido, el número de capas del *G-code* y las estimaciones del *slicer*. Al incorporar la comparación geométrica con el modelo de referencia, el progreso se expresa como una medida objetiva e interpretable del estado de la pieza, y no únicamente como una predicción temporal. El sistema se ha diseñado como un algoritmo modular que integra adquisición, preprocesamiento, registro y evaluación, lo que facilita su adaptación a diferentes escenarios y geometrías.

La adquisición se apoya en un sensor RGB-D para reconstruir nubes de puntos a partir de imágenes de color y profundidad, mientras que el registro robusto entre la nube observada y el modelo teórico se apoya en algoritmos especializados. Esta arquitectura permite su integración tanto en configuraciones experimentales de laboratorio como en celdas de fabricación, donde el sensor puede colocarse en diferentes posiciones y normalizar las capturas mediante referencias geométricas en la escena.

Finalmente, el enfoque se aplica también como herramienta de validación, ya que el análisis conjunto de métricas permite interpretar el estado de la impresión y discriminar el grado de similitud entre la pieza impresa y el modelo teórico, habilitando criterios para detección temprana de desviaciones y soporte a la decisión operativa.

B.2. Evaluación de impactos

La evaluación de impactos se organiza en cuatro categorías, social, económico, medioambiental y otros impactos relevantes para el tipo de desarrollo realizado. En cada caso se describen beneficios esperables, riesgos y medidas de mitigación para un uso responsable.

B.2.1. Impacto social

El impacto social del sistema se asocia a cambios en la forma de supervisar la impresión y en la interacción del operador con el proceso. En cuanto a los beneficios, al detectar desviaciones durante la fabricación se reduce la probabilidad de completar piezas defectuosas y, con ello, la exposición a retrabajos. A causa de esto se mejora la experiencia del operador y la seguridad operativa. Además, las salidas visuales y métricas facilitan la comprensión del proceso, la evaluación de configuraciones de impresión y la formación de personal técnico. Por último, la exportación de métricas y evidencias visuales favorece la trazabilidad, la transparencia y la realización de auditorías internas y documentación del proceso.

En cuanto a los riesgos, una dependencia excesiva del sistema, especialmente si existe una calibración inadecuada o umbrales mal ajustados, podría conducir a decisiones incorrectas, como detener una impresión válida o continuar una impresión defectuosa.

Finalmente, para un uso responsable se recomienda mantener un enfoque en el que el sistema recomiende y el operador confirme acciones. Esto preserva la responsabilidad humana en la validación, incorporar umbrales conservadores y alertas graduadas con explicación de las métricas que motivan la advertencia..

B.2.2. Impacto económico

El impacto económico se relaciona con costes evitados por fallos, con mejoras de productividad y con inversiones necesarias para su adopción. En cuanto a los beneficios, la detección temprana puede evitar consumir material y tiempo de máquina en impresiones, reduciendo costes por desperdicio. Por otro lado, la introducción de trazabilidad y métricas comparables facilita el ajuste sistemático de parámetros y mejora la repetibilidad de impresiones, incrementando productividad. Finalmente, una arquitectura modular favorece la reutilización en distintos modelos y geometrías, reduciendo el coste marginal de adopción.

En cuanto a los costes de barrea, la sensorización y el cómputo implican inversión, mantenimiento y calibración, y pueden introducir tiempos de parada por verificación. Adicionalmente, la integración en una celda industrial puede requerir adecuaciones en seguridad, red y trazabilidad, aumentando el coste de despliegue.

Para mitigar estas barreras, se recomienda definir niveles de despliegue, modo laboratorio, modo preproducción y modo producción, con requisitos crecientes y objetivos verificables. Además, es importante documentar procedimientos de calibración y criterios de aceptación para reducir la incertidumbre operativa y los costes de explotación.

B.2.3. Impacto medioambiental

El impacto medioambiental combina efectos positivos a causa de la reducción de desperdicio con efectos asociados al consumo energético y a la incorporación de hardware. Al evitar finalizar piezas defectuosas, se disminuye el consumo de filamento y la energía de máquina, reduciendo residuos. Además, disponer de métricas objetivas acelera la convergencia hacia parámetros de proceso estables, reduciendo iteraciones fallidas y, por tanto, material y energía consumidos en pruebas.

Por otro lado, la adquisición con sensores, la iluminación auxiliar y el cómputo incrementan el consumo energético frente a una impresión no monitorizada, y la incorporación de dispositivos implica impactos asociados a la fabricación y al fin de vida del equipamiento. Para ello se recomienda operar con estrategias de adquisición eficientes, como capturas bajo demanda y reducción de tasa de muestreo cuando el proceso sea estable. Además, se debe de priorizar hardware modular y reparable, incorporando planes de reutilización o reciclaje cuando aplique.

B.2.4. Otros impactos relevantes

Además de los impactos anteriores, el trabajo presenta impactos científico-técnicos y organizativos. En el primer caso se aporta una metodología reproducible de adquisición, normalización, registro y evaluación. También facilita la extensión a nuevos sensores, nuevas métricas o nuevos métodos de registro debido a la separación en módulos. En cuanto a lo organizativo, el sistema favorece una cultura de calidad basada en datos, sustituyendo decisiones informales por indicadores medibles y exportables, lo que puede mejorar la coherencia de criterios y la comparabilidad entre ejecuciones.

B.3. Análisis de aspectos éticos

El desarrollo y el potencial despliegue del sistema implica consideraciones éticas asociadas a privacidad, seguridad, fiabilidad y uso responsable de resultados.

B.3.1. Privacidad y protección de datos

El sistema emplea un sensor RGB-D y, por tanto, puede capturar información visual del entorno. En escenarios de laboratorio o fábrica podrían aparecer personas o elementos identificables, por lo que debe aplicarse el principio de minimización capturando únicamente la región relevante de impresión. También se debe de evitar almacenar imágenes crudas y priorizar nubes de puntos ya recortadas. Finalmente, es importante definir políticas de retención, acceso y control de los datos exportados, especialmente si se guardan imágenes, registros de ejecución o ficheros asociados a la trazabilidad del proceso.

B.3.2. Fiabilidad, riesgo de error y responsabilidad

Las métricas del sistema pueden influir en decisiones como detener una impresión, declarar una pieza inválida o aceptar un resultado. Éticamente, resulta esencial comunicar limitaciones y condiciones de validez, incluyendo ruido del sensor, dependencia del preprocesamiento y sensibilidad a umbrales. Además, se debe de evitar el uso del sistema como único criterio de calidad sin un procedimiento de contraste. Para sostener la responsabilidad, debe mantenerse trazabilidad registrando parámetros de captura, umbrales, configuración y versión del software, de manera que los resultados puedan interpretarse y auditarse.

B.3.3. Transparencia, interpretabilidad y trazabilidad

El sistema incorpora métricas y visualizaciones que mejoran la interpretabilidad, pero su uso responsable requiere acompañar las métricas con descripciones legibles, unidades coherentes y documentación clara del significado operativo de cada indicador. Del mismo modo, conviene explicar la relación entre métrica y decisión, definiendo criterios de alerta y aceptación, e incluyendo ejemplos de casos típicos de fallo, con el fin de reducir interpretaciones erróneas y facilitar la supervisión humana.

B.3.4. Propiedad intelectual y licencias

El desarrollo integra bibliotecas externas, por lo que es necesario respetar licencias de software y atribuciones requeridas. También se debe atender restricciones de uso si se incorpora software o algoritmos con condiciones específicas, y gestionar adecuadamente la propiedad intelectual del código propio si se distribuye en contextos distintos al académico. Esto con el objetivo de garantizar que la difusión del trabajo no incumpla obligaciones legales ni éticas asociadas a licencias.

B.4. Contribución a los Objetivos de Desarrollo Sostenible

El sistema propuesto, al centrarse en la monitorización geométrica y la validación del proceso de impresión 3D, contribuye a la eficiencia productiva y a la incorporación de tecnologías de inspección avanzadas en entornos de fabricación. En particular, el trabajo se alinea con los siguientes Objetivos de Desarrollo Sostenible:

- **ODS 8: Trabajo decente y crecimiento económico.** La mejora de fiabilidad y eficiencia del proceso puede traducirse en reducciones de costes operativos asociados a reprocesos y rechazo de piezas, aumentando productividad y competitividad. La adopción de inspección basada en visión 3D y análisis de datos también impulsa perfiles técnicos especializados.
- **ODS 9: Industria, innovación e infraestructura.** El desarrollo de un proceso basado en adquisición 3D, registro robusto y métricas cuantitativas de progreso y similitud contribuye a la digitalización del control de calidad en fabricación aditiva. Promueve procesos automatizados, trazables y reproducibles, facilitando la producción inteligente.

- **ODS 12: Producción y consumo responsables.** La capacidad de evaluar la impresión durante su ejecución permite detectar desviaciones tempranas y reducir la finalización de piezas defectuosas. Esto disminuye reimpresiones, desperdicio de material y tiempo de máquina, favoreciendo un uso más eficiente de recursos.

B.5. Marco normativo de aplicación

El sistema desarrollado es un prototipo software-hardware de monitorización para un proceso industrial o de laboratorio. Por ello, su marco normativo depende del contexto de despliegue que puede ser investigación, laboratorio, preproducción o producto comercial.

Cuando el sistema se integra en una celda de fabricación, conviene considerar la normativa de seguridad aplicable a maquinaria y puestos de trabajo. En ese caso, debe realizarse una evaluación de riesgos del conjunto, incluyendo impresora, sensor, soporte mecánico y cableado, y se debe de contemplar riesgos eléctricos, mecánicos y de atrapamiento.

También deben definirse señalización y procedimientos de operación segura. Esto es especialmente relevante si existen movimientos del extrusor o del conjunto cámara en la zona de trabajo.

En contextos donde se capturen imágenes o datos que puedan identificar personas, debe considerarse la normativa de protección de datos aplicable. En particular, deben respetarse principios de retención y seguridad. En la práctica, esto implica minimizar captura no necesaria mediante recorte y control de almacenamiento, definir roles y permisos de acceso a ficheros exportados e incorporar medidas básicas de seguridad en almacenamiento y transferencia.

Si el sistema evoluciona hacia un producto integrable en entorno industrial, pueden aplicarse requisitos técnicos adicionales. Entre ellos destacan la seguridad eléctrica del equipamiento y de las fuentes de alimentación. También es relevante la compatibilidad electromagnética si se conecta a redes o a equipos de control. Además, pueden existir requisitos internos del laboratorio u organización que suelen afectar a instalación, cableado y operación de sensores y dispositivos.

Cuando el sistema se use como apoyo a decisiones de calidad, conviene alinearlos con procedimientos internos de aseguramiento de calidad. Para ello, es importante registrar parámetros de ejecución, configuración de cámara, umbrales de métricas y versiones de software. Además, conviene definir procedimientos de calibración y verificación periódica. Por último, es recomendable generar evidencias exportables, como CSV de métricas, figuras y registros de ejecución.

B.6. Estándares aplicables y contribuciones

Además del marco normativo, existen estándares técnicos y buenas prácticas que afectan al desarrollo y despliegue del sistema. En fabricación aditiva es habitual apoyarse en familias de estándares. Estas cubren terminología, principios generales, requisitos de diseño y aspectos de calidad. En muchos casos se trata de normativas conjuntas ISO y ASTM.

En el contexto de este trabajo es importante asegurar consistencia en las definiciones. Esto incluye conceptos relacionados con proceso, defectos y métricas. También es relevante encuadrar el proceso FDM y sus variabilidades, además de describir de forma rigurosa sistemas de coordenadas y referencias. Esto permite garantizar comparabilidad entre mediciones.

El trabajo se alinea con estas prácticas al definir explícitamente marcos de referencia, normalización geométrica y fija unidades y métricas reproducibles.

Desde la perspectiva de metrología y digitalización 3D, la inspección dimensional suele requerir guías de verificación y calibración. Por ello, son relevantes la consistencia de intrínsecos y la estabilidad temporal del sensor. También lo son la repetibilidad y la trazabilidad de la medición cuando se requiera cuantificación metrológica. Además, conviene identificar fuentes de incertidumbre como el ruido de profundidad, las oclusiones, la discretización por vóxeles y la dependencia del preprocesamiento.

Como contribución práctica, este trabajo deja definidos puntos de control operativos como el recorte coherente y la normalización mediante referencias. También se incluye la exportación de métricas para análisis repetible.

Si el sistema se industrializa, resultan relevantes estándares y guías vinculados a gestión de calidad y ciclo de vida del software. Esto incluye control documental, validación y verificación, y trazabilidad de requisitos. También incluye mantenibilidad, reproducibilidad, pruebas y control de versiones. La arquitectura modular del pipeline facilita estos objetivos, así como la separación por componentes favorece la mantenibilidad, pruebas unitarias por módulo y permite una extensión controlada.

En despliegues con almacenamiento en red o exportación automática de datos, cobran importancia prácticas de seguridad de la información, esto también aplica si existe integración con infraestructura industrial. En estos casos conviene considerar control de accesos y permisos, registro de actividad y la trazabilidad de modificaciones. Además, debe protegerse la información exportada y planificar copias de seguridad.

Aunque el objetivo principal del trabajo no es proponer un estándar, se aportan elementos reutilizables que contribuyen a buenas prácticas. Se incluye la definición ope-

racional de métricas de solapamiento, cobertura superficial y progreso en altura. Estas métricas se plantean con umbrales geométricos claros y se propone un procedimiento de normalización geométrica basado en referencias en escena para lograr comparabilidad entre capturas. Además, se aporta una estructura modular que facilita auditoría, pruebas y sustitución de componentes.

B.7. Síntesis

El sistema propuesto habilita monitorización geométrica durante la impresión. Esto puede generar impactos positivos en eficiencia, reducción de desperdicio y trazabilidad.

Para un despliegue responsable es necesario considerar riesgos sociales, económicos y medioambientales. También deben considerarse aspectos éticos asociados a privacidad y fiabilidad.

El marco normativo y de estándares relevante depende del contexto de aplicación. Sin embargo, puede abordarse de forma ordenada, para ello se requiere evaluación de riesgos, protección de datos y trazabilidad de resultados. También es importante adoptar buenas prácticas de metrología y calidad del software.

Planificación temporal y presupuesto

C.1. Estructura de Paquetes de Trabajo

En la Figura C.1 se presenta un resumen de las fases del proyecto y las tareas claves orientadas al diseño, implementación y validación de un sistema de adquisición y análisis de nubes de puntos con sensor RGB-D.

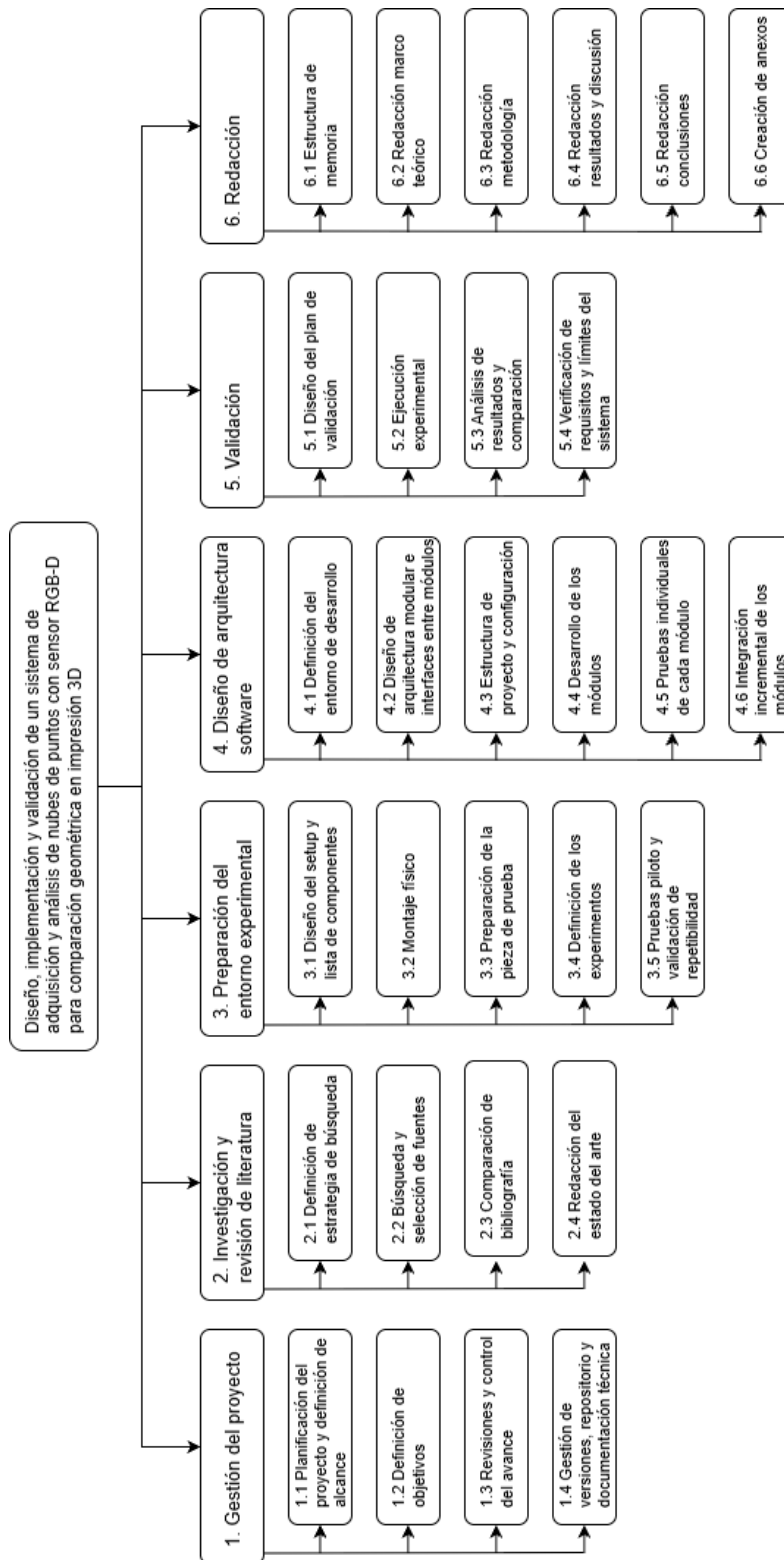


Figura C.1 Paquetes de trabajo del proyecto

C.2. Planificación

En esta sección se detalla la planificación temporal del proyecto correspondiente a los paquetes de trabajo presentados en la Figura C.1. Para ello se ha utilizado un Diagrama de Gantt en la Figura C.2.

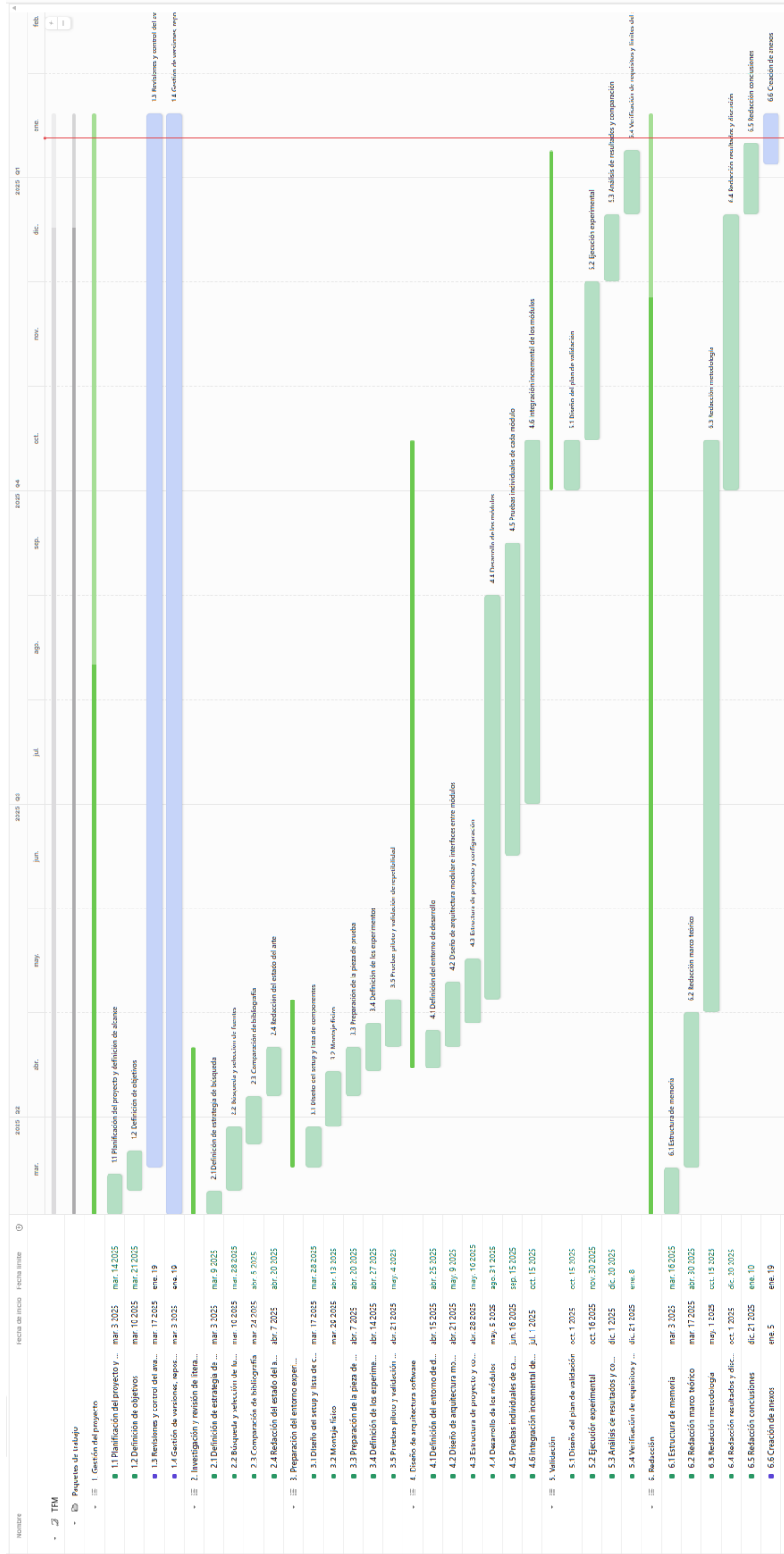


Figura C.2 Diagrama de Gantt del proyecto

C.3. Presupuesto

En esta sección se presentan los recursos financieros utilizados en el proyecto. Se especifican los costes asociados a materiales, equipos y personal del proyecto.

C.3.1. Costo de los materiales

Acá se detallan los costes asociados a los materiales necesarios para el desarrollo del proyecto. Estos incluyen los componentes hardware, consumibles y elementos auxiliares requeridos para la implementación del sistema de adquisición y procesamiento de nubes de puntos. La Tabla C.1 recoge el coste unitario y total de cada material empleado.

Concepto	Unidades	Costo por unidad	Costo total
Brazo articulado 4 GDL con mordaza	1	30,00 €	30,00 €
Cable USB-C / USB 3.0 para cámara	1	10,00 €	10,00 €
Marcadores ArUco	4	1,50 €	6,00 €
Filamento PLA/PETG (bobina 1 kg)	2	25,00 €	50,00 €
SSD externo 1 TB	1	80,00 €	80,00 €
Consumibles varios (cinta doble cara, pegamento, bridas, etc.)	1	10,00 €	10,00 €
Total			186,00 €

Tabla C.1 Costo de los materiales del proyecto

C.3.2. Amortización de los equipos

Dado que algunos de los equipos utilizados tienen una vida útil superior a la duración del proyecto, se ha considerado su amortización parcial. En la Tabla C.2 se presenta el cálculo de la amortización de los equipos, teniendo en cuenta su coste de adquisición y el periodo de uso dentro del proyecto.

Concepto	Costo total	Años	Costo de amortización / año	Uso (año)	Costo de amortización
Prusa MK3 3D Printer	999,00 €	5,00	199,80 €	1,00	199,80 €
Laptop	1.250,00 €	5,00	250,00 €	1,00	250,00 €
Cámara RGB-D Intel RealSense	400,00 €	5,00	80,00 €	1,00	80,00 €
Costo total de amortización					529,80 €

Tabla C.2 Amortización de los equipos

C.3.3. Costo del personal

En cuanto al coste asociado al personal implicado en el desarrollo del proyecto, este se ha estimado en función del tiempo dedicado a cada tarea y de una tarifa horaria representativa. La Tabla C.3 resume el coste total del personal necesario para la ejecución del proyecto.

Concepto	Costo de hora	Horas	Costo
Director del proyecto	40,00 €	40	1.600,00 €
Investigador predoctoral	20,00 €	120	2.400,00 €
Autor del proyecto	7,00 €	500	3.500,00 €
Costo total del personal			7.500,00 €

Tabla C.3 Costo del personal del proyecto

C.3.4. Costo total del proyecto

Finalmente, en la Tabla C.4 se presenta el costo total del proyecto según los apartados anteriores.

Concepto	Costo
Costo de los materiales	186,00 €
Amortización de los equipos	529,80 €
Costo del personal	7.500,00 €
Costo total del proyecto	8.215,80 €

Tabla C.4 Costo total del proyecto