

DEPARTAMENTO DE INFORMÁTICA

Universidad Carlos III de Madrid



Proyecto fin de carrera

TÉCNICAS Y HERRAMIENTAS DE PROCESAMIENTO DE NUBES DE PUNTOS TRIDIMENSIONALES

Autor: Rubén Ares Cano
Director: Miguel Ángel Patricio Guisado

En Colmenarejo, a 27 de Octubre de 2015

Título: Técnicas y herramientas de procesamiento de nubes de puntos tridimensionales
Autor: Rubén Ares Cano
Director: Miguel Ángel Patricio Guisado

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de 20__
en Colmenarejo, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid,
acuerda otorgarle la CALIFICACIÓN de

PRESIDENTE

VOCAL

SECRETARIO

AGRADECIMIENTOS

Agradezco, en primer lugar, a Miguel Ángel Patricio Guisado la oportunidad de realizar este proyecto.

En segundo lugar, agradecer al Grupo de Inteligencia Artificial Aplicada (GIAA) por su apoyo a la hora de realizar el proyecto proporcionándome todo el material necesario y toda ayuda que he podido necesitar.

Por último, agradecer a mi familia sus esfuerzos y ánimos para que yo pudiera completar mis estudios.

RESUMEN

Este trabajo se divide fundamentalmente en tres bloques.

En primer lugar, se realiza un análisis de las diferentes técnicas de captura y procesamiento de nubes de puntos de forma que se establece un marco teórico al proyecto y sienta las bases del trabajo posterior.

En segundo lugar, se realiza un análisis de MeshLab y Point Cloud Library, dos de las herramientas software disponibles a nivel general más importantes para trabajar con nubes de puntos y modelos tridimensionales, ambas Open Source.

Por último, contando con el dispositivo de captura SR4000 de MESA Imaging, se realizan diversas pruebas de concepto utilizando las herramientas y técnicas anteriormente descritas. Parte de las pruebas de concepto descritas en este trabajo están orientadas hacia el ámbito de la antropometría debido a la colaboración realizada con Manuel Sillero, profesor titular en la Facultad de Ciencias de la Actividad Física y del Deporte (INEF-Madrid) y con José Manuel Maestre Rodríguez, alumno de master en la misma y cuyo trabajo de fin de master está relacionado con la cámara SR4000 utilizada en este trabajo así como con el contenido del mismo en general.

PALABRAS CLAVE. 3D tridimensional captura procesamiento nube puntos MeshLab PCL TOF

ABSTRACT

This work is mainly divided into three blocks.

First, an analysis of the different techniques of capturing and processing point cloud so that a theoretical framework for the project is established and provides the basis for further work is done.

Second, an analysis of MeshLab and Cloud Point Library, two of the available software tools to work with most important point clouds and three-dimensional models, two Open Source general level is performed.

Finally, with the capture device SR4000 MESA Imaging, various concept tests are performed using the tools and techniques described above. Part of the proof of concept described in this paper are oriented towards the field of anthropometry due to the collaboration made with Manuel Sillero, professor at the Faculty of Physical Activity and Sport (INEF-Madrid) and José Manuel Maestre Rodriguez, a student of master in it and whose work is related to master the SR4000 camera used in this work as well as the contents in general.

KEYWORDS. 3D tridimensional capture processing cloud point MeshLab PCL TOF

Índice de contenido

CAPÍTULO 1. INTRODUCCIÓN	15
1.1 Introducción al proyecto	16
1.2 Objetivos del proyecto	18
1.3 Material utilizado	20
1.4 Contenido del documento	21
CAPÍTULO 2. PRECEDENTES.....	23
2.1 Proyecto Fin de Carrera de Daniel Sánchez del Álamo	24
2.1.1 Objetivo principal del proyecto de Daniel.....	24
2.1.2 El API de la cámara y la librería SwissRanger	25
2.2 Trabajos de José Manuel Maestre	26
CAPÍTULO 3. ESTADO DEL ARTE	29
3.1 Captura de datos tridimensionales	30
3.1.1 Métodos de contacto	30
3.1.2 Métodos pasivos sin contacto.....	30
3.1.3 Métodos activos sin contacto	31
3.2 Procesamiento de datos tridimensionales.....	34
3.2.1 Filtrado de datos	34
3.2.2 Suavizado de superficies	34
3.2.3 Alineado	34
3.2.4 Segmentado	35
3.3 Modelización de los datos tridimensionales.....	36
3.3.1 Modelos de malla poligonal	36
3.3.2 Modelos de superficies	36
3.3.3 Modelos sólidos CAD.....	37
3.4 Aplicaciones del escaneado tridimensional	38
CAPÍTULO 4. SOFTWARE PARA EL PROCESADO TRIDIMENSIONAL.....	41
4.1 Software desarrollado.....	42
4.1.1 Análisis previo	42
4.1.2 Formatos de almacenamiento utilizados.....	42
4.1.3 Herramientas de filtrado	45
4.2 MeshLab.....	52
4.2.1 Descripción.....	52
4.2.2 Plataformas soportadas	53
4.2.3 Licencia	53
4.2.4 Instalación	53
4.2.5 Funcionamiento	54
4.3 PCL (Point Cloud Library).....	59
4.3.1 Descripción.....	59
4.3.2 Plataformas soportadas	62
4.3.3 Licencia	62
4.3.4 Instalación	62
4.3.5 Funcionamiento	63
CAPÍTULO 5. PRUEBAS DE CONCEPTO	69
5.1 Capturas realizadas	70

5.1.1 Desarrollo de las capturas	70
5.1.2 Prueba inicial	70
5.1.3 Captura del maniquí completo.....	73
5.1.4 Captura del torso del maniquí.....	77
5.1.5 Captura de Manuel Sillero.....	79
5.2 Procesos con PCL.....	83
5.2.1 Estado inicial.....	83
5.2.2 Filtrado de la nube de puntos	83
5.2.3 Suavizado de la nube de puntos.....	84
5.2.4 Triangulación de la nube de puntos.....	85
5.3 Procesos con MeshLab	87
5.3.1 Alineado manual de nubes de puntos.....	87
5.3.2 Triangulación de un rostro	90
5.4 Otros procesos de interés	92
5.4.1 Suavizado mediante técnicas estadísticas básicas	92
5.4.2 Alineación automática de varias nubes de puntos	94
CAPÍTULO 6. CONCLUSIONES	97
6.1 La cámara SR4000	98
6.2 Software desarrollado.....	99
6.3 MeshLab	100
6.4 Point Clouds Library (PCL)	102
6.5 Antropometría.....	104
6.5 Conclusiones finales	105
ANEXOS	107
ANEXO I. Presupuesto del proyecto.....	108
ANEXO II. La cámara SR4000.....	111
ANEXO III. Introducción a la antropometría.....	116
ANEXO IV. Proyecto de escaneo tridimensional real	124
ANEXO V. Código fuente de las pruebas	126
Bibliografía	129

Índice de ilustraciones

Ilustración 1. Escaneado tridimensional.	16
Ilustración 2. Manuel y José Manuel posando junto al maniquí.	18
Ilustración 3. La tecnología TOF de la cámara SR4000 orientada al ocio.	24
Ilustración 4. Método de triangulación.....	31
Ilustración 5. Modelo de malla poligonal.....	36
Ilustración 6. Ejemplo de curva NURBS.....	36
Ilustración 7. Escaneado del David de Miguel Ángel.	38
Ilustración 8. Imagen previa al filtrado	46
Ilustración 9. Imagen de ejemplo filtrando la confiabilidad al 80%.....	46
Ilustración 10. Imagen de ejemplo filtrando la distancia.....	47
Ilustración 11. Imagen de ejemplo con filtro rectangular.....	48
Ilustración 12. Imagen de ejemplo previa al filtro de puntos aislados	49
Ilustración 13. Imagen de ejemplo posterior al filtro de puntos aislados	49
Ilustración 14. Utilización conjunta de los filtros.....	51
Ilustración 15. Logotipo de MeshLab.	52
Ilustración 16. Visión general de MeshLab.	54
Ilustración 17. Pestaña File en MeshLab.....	55
Ilustración 18. Vista de capas.....	55
Ilustración 19. Pestaña Edit en MeshLab.	56
Ilustración 20. Pestaña Filters en MeshLab.....	56
Ilustración 21. Ejemplo de ayuda en MeshLab.	58
Ilustración 22. Pestaña Render en MeshLab.....	58
Ilustración 23. Logotipo de PCL.....	59
Ilustración 24. Estructura de PCL.	59
Ilustración 25. Ejemplo de filtrado con PCL.	60
Ilustración 26. Ejemplo de árbol-kd.	61
Ilustración 27. Ejemplo de Sample_consensus en PCL.	61
Ilustración 28. Crear un proyecto de PCL con Visual Studio 2010.	64
Ilustración 29. Ruta hacia las carpetas include.	65
Ilustración 30. Incluir librerías en PCL.	66
Ilustración 31. Incluir librerías exactas PCL.....	67
Ilustración 32. Salida del programa de ejemplo de PCL.....	68
Ilustración 33. Captura de la prueba inicial.....	70
Ilustración 34. Visualización de la prueba inicial sin filtrado.	71
Ilustración 35. Error capturado.	71
Ilustración 36. Prueba inicial. Aplicación del filtro rectangular.	72
Ilustración 37. Prueba inicial. Aplicación del filtro de confianza.	72
Ilustración 38. Prueba inicial. Visualización final en MeshLab.....	73
Ilustración 39. Prueba inicial. Visualización final en PCL.....	73
Ilustración 40. Segunda prueba de concepto. Primer caso.....	74
Ilustración 41. Secuencia A filtrada.....	75
Ilustración 42. Nube de puntos en el caso A.....	76
Ilustración 43. Torso del maniquí.....	77
Ilustración 44. Secuencia B filtrada.	78

Ilustración 45. Nube de puntos del torso del maniquí.....	79
Ilustración 46. Manuel Sillero posando para la tercera prueba.....	80
Ilustración 47. Secuencia de Manuel Sillero filtrada.....	81
Ilustración 48. Fallo en la captura del torso.....	82
Ilustración 49. Estado inicial en la prueba con PCL.....	83
Ilustración 50. Modelo de la prueba con PCL filtrado.....	84
Ilustración 51. Modelo de la prueba con PCL suavizado.....	85
Ilustración 52. Modelo de la prueba con PCL triangulado.....	86
Ilustración 53. Captura inicial para las pruebas con MeshLab.....	87
Ilustración 54. Selección de puntos para alinear.....	88
Ilustración 55. Estado inicial de las nubes de puntos.....	88
Ilustración 56. Estado final, perspectiva 1.....	89
Ilustración 57. Estado final, perspectiva 2.....	89
Ilustración 58. Triangulación de un rostro con MeshLab.....	90
Ilustración 59. Superficie del rostro triangulado.....	90
Ilustración 60. Fallo de captura de superficie.....	92
Ilustración 61. Suavizado utilizando varios frames.....	93
Ilustración 62. Segunda captura suavizada.....	94
Ilustración 63. Cámara TOF SR4000.....	111
Ilustración 64. Frontal de la cámara SR4000.....	112
Ilustración 65. Imágen de distancia.....	113
Ilustración 66. Imágen de amplitud.....	114
Ilustración 67. Imágen de confianza.....	115
Ilustración 68. Ejemplo de antropometría.....	116
Ilustración 69. Referencias antropométricas generales.....	118
Ilustración 70. Fotografía de un segmómetro.....	119
Ilustración 71. Escaneado tridimensional de una persona.....	120
Ilustración 72. Modelo tridimensional de una persona generado por ordenador.....	120
Ilustración 73. Toma de datos antropométricos mediante escaneado laser y plataforma.....	121
Ilustración 74. Puente romano de Vinuesa.....	124
Ilustración 75. Escaner laser HDS-3000 de Leica.....	125
Ilustración 76. Resultados de un proyecto real.....	125

Índice de tablas

Tabla 1. Hardware utilizado en el proyecto.	20
Tabla 2. Software utilizado en el proyecto.	20
Tabla 3. Formato de fichero TSF	43
Tabla 4. Formato de fichero ASC.....	44
Tabla 5. Formato de fichero PCD	45
Tabla 6. Rutas a incluir en PCL.....	65
Tabla 7. Rutas a las librerías de PCL.	66
Tabla 8. Librerías incluidas para el ejemplo PCL.	67
Tabla 9. Precio hardware del presente proyecto.....	108
Tabla 10. Precio software del presente proyecto.....	108
Tabla 11. Precio del personal del presente proyecto.	108

CAPÍTULO 1. INTRODUCCIÓN

En este capítulo se realiza una introducción al proyecto realizado. En primer lugar, se propone un marco teórico en el que se asienta el proyecto. Posteriormente, se definen los diferentes objetivos propuestos a la hora de realizar dicho proyecto. En tercer lugar, se describe el material que ha sido necesario para llevar a cabo el proyecto. Para finalizar, se describe estructuralmente el documento y su contenido.

CAPÍTULO 1

1.1 INTRODUCCIÓN AL PROYECTO

1.2 OBJETIVOS DEL PROYECTO

1.3 MATERIAL UTILIZADO

1.4 CONTENIDO DEL DOCUMENTO

1.1 Introducción al proyecto

Los sistemas de adquisición de imágenes tridimensionales se encuentran experimentando un gran auge en la actualidad. Hasta hace poco tiempo, su excesivo precio los hacía inalcanzables salvo para proyectos de gran magnitud y grandes presupuestos. Debido a esto, sus aplicaciones prácticas estaban muy limitadas. Sin embargo, la mejora constante de la tecnología y la reducción de costes asociada han propiciado que poco a poco se hayan ido integrando en aplicaciones más cotidianas, por lo que en la actualidad va siendo cada vez más habitual encontrar este tipo de tecnología.

La primera aplicación práctica del escaneado tridimensional surgió en la industria como forma de control de las dimensiones en piezas con pequeños márgenes de tolerancia. A partir de esto, ha ido expandiéndose hacia otros campos como la medicina o la generación de modelos de terrenos. Actualmente, se puede considerar que la industria del ocio es una de las grandes impulsoras de esta tecnología dado que es cada vez es más habitual encontrar dispositivos domésticos que cuentan con algún tipo de elemento de visión tridimensional y que ya no son solamente utilizados como método de obtención de datos sino como herramienta de interacción. Un ejemplo es el de las consolas de última generación que cuentan en la mayoría de los casos con algún tipo de sistema de visión tridimensional y basta con el movimiento del cuerpo para enviar órdenes al dispositivo.

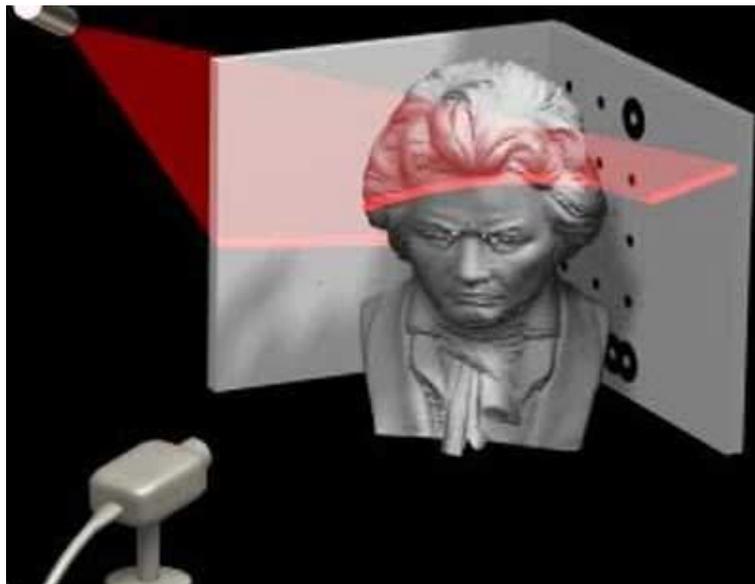


Ilustración 1. Escaneado tridimensional.

Por otra parte, el desarrollo de la aplicación práctica de una tecnología en particular viene dado generalmente por el desarrollo de las dos patas que componen cualquier tecnología informática: el hardware y el software.

En el caso de la visión o escaneado tridimensional, desarrollar el hardware implica el hecho de producir nuevos dispositivos de captura mejores y más baratos. En este sentido, existen en la actualidad multitud de sistemas de escaneado tridimensional, cada uno con sus características propias, sus ventajas y sus inconvenientes. Esta diversidad genera la posibilidad de elegir un sistema u otro en función de diversas variables a tener en cuenta para cada proyecto como pueden ser el coste, la precisión, el entorno o la portabilidad del sistema, por poner algunos

ejemplos. Uno de los puntos fundamentales para permitir el desarrollo de esta tecnología ha sido el descenso en el coste de los dispositivos de captura que han pasado de costar varios miles de euros a solamente algunos cientos.

Por su parte, al hablar de software en el desarrollo de una tecnología no hay que centrarse en la aplicación final sino en el conjunto de herramientas que facilitan el trabajo con la misma, tales como frameworks y librerías de desarrollo. Este tipo de herramientas facilitan la proliferación del software final, hecho que impulsa el desarrollo de la tecnología. En los inicios del escaneado tridimensional e incluso aún en la actualidad, era normal encontrar herramientas específicas de cada fabricante para sus propios dispositivos, en la mayoría de los casos cerradas, disponibles solo para aquellos que adquirieran un escáner de su marca y con funcionalidad limitada. Este hecho suponía un lastre en el desarrollo de aplicaciones finales debido a la ingente cantidad de trabajo necesaria para desarrollar las comunicaciones con el dispositivo así como los diferentes algoritmos de tratamiento de datos requeridos para la aplicación. Por suerte, en la actualidad han surgido diversas herramientas que, con mayor o menor funcionalidad, están disponibles para el desarrollador y proveen de múltiples algoritmos y procesos que poder aplicar en la aplicación a desarrollar. Por otra parte, cada vez con más frecuencia, los dispositivos ofrecen alguna librería de conexión o SDK con el que poder integrar el escáner directamente en el proyecto sin necesidad de utilizar las aplicaciones propias de la marca, cosa que facilita el desarrollo de nuevas aplicaciones.

1.2 Objetivos del proyecto

El objetivo global del proyecto descrito en este documento consiste en realizar un estudio general del estado del arte en el ámbito de las herramientas y técnicas de procesado de datos tridimensionales disponibles en la actualidad sin olvidar realizar un repaso de las técnicas de captura, así como realizar un examen detallado de dos de las herramientas mencionadas y llevar a cabo diversas pruebas de concepto con las mismas.

Es necesario mencionar que durante el desarrollo de las pruebas de concepto de este proyecto surgió la posibilidad de colaborar con Manuel Sillero, profesor titular en la Facultad de Ciencias de la Actividad Física y del Deporte (INEF-Madrid) y con José Manuel Maestre Rodríguez, alumno de master en la misma. Parte de los trabajos de José Manuel en el master tienen como objetivo el estudio de la viabilidad de la tecnología TOF utilizando la cámara SR4000 aplicada a diversos estudios



Ilustración 2. Manuel y José Manuel posando junto al maniquí.

antropométricos. Debido a la afinidad entre ambos proyectos, las pruebas de concepto desarrolladas para este proyecto se dirigieron hacia los objetivos propuestos por José Manuel pues presentaban todos los elementos necesarios para ilustrar este estudio (captura y procesamiento de datos tridimensionales) y aportaban una aplicación práctica a las pruebas, lo que las dota de mayor interés.

Para alcanzar el objetivo global del proyecto, se divide el trabajo en cinco bloques principales:

- En primer lugar, se considera necesario realizar un repaso de los elementos de partida con los que se cuenta al comenzar el proyecto. Como precedente principal, Daniel Sánchez del Álamo Benguigui en su proyecto fin de carrera “Time-Of-Flight Camera” (1) realiza un profundo estudio de las capacidades de la cámara SR4000 y desarrolla una librería que permite operar con ella de forma cómoda y sencilla. Dado que dicha cámara es la utilizada para realizar las pruebas de concepto de este proyecto, parece evidente la utilidad de su trabajo. Por otra parte, como se ha comentado anteriormente, la colaboración con José Manuel Maestre durante el proyecto dirige las pruebas de concepto de este proyecto por lo que se considera interesante realizar una pequeña aproximación a sus trabajos anteriores basados en la tecnología TOF así como a los objetivos que propone en su trabajo actual.
- En segundo lugar se realiza un estudio teórico del estado del arte de las técnicas de escaneado y procesamiento tridimensional más habituales así como sus principales aplicaciones prácticas en la actualidad. De este modo se consigue proponer un marco teórico al proyecto que sirve de guía en el desarrollo del resto del trabajo a realizar.

- En tercer lugar se afronta el estudio de las herramientas actuales disponibles para el procesamiento de datos tridimensionales. Si bien en el documento se hace mención a varias herramientas, el estudio se centra en dos de ellas, MeshLab (2) y PCL (3). La elección de estas dos herramientas viene dada por el hecho de que ambas pertenecen a proyectos de código abierto y están disponibles para el público en general. Por otro lado, se consideran más completas que cualquier otra de características similares que se ha encontrado. Por último, el hecho de estudiar las dos en lugar de solo una viene dado por el hecho de que abarcan conceptos diferentes. MeshLab puede considerarse una herramienta de trabajo en si misma mientras que PCL es una librería que podemos integrar en cualquier proyecto en el que se necesite trabajar con nubes de puntos.
- En cuarto lugar, analizadas las herramientas y técnicas de procesamiento de datos tridimensionales, se describen varias pruebas de concepto realizadas con la intención de examinar las herramientas y técnicas ya mencionadas. Las pruebas de concepto realizadas están basadas en el uso de la cámara de escaneo tridimensional con tecnología Time Of Flight (en adelante TOF) SR4000 de MESA Imaging (4). Para realizar dichas pruebas, se comenzó con el desarrollo previo de un software que permitiese realizar un filtrado previo de las capturas realizadas con la cámara disponible y permitiese utilizar los datos obtenidos con las herramientas software analizadas en este trabajo.
- Para finalizar, se establecen las conclusiones a las que se ha llegado tras la realización del proyecto respecto a todos los elementos que han participado en el proyecto, es decir, la cámara, Meshlab, PCL, etc.

1.3 Material utilizado

Para el desarrollo de este proyecto no ha sido necesario utilizar una gran cantidad de elementos hardware o software más allá de los básicos. A continuación se describen los elementos hardware y software cuya importancia ha sido capital para la realización del proyecto.

Los elementos hardware requeridos para desarrollar el proyecto han sido los siguientes:

Elemento	Descripción
Ordenador personal	El material básico tanto a la hora de redactar este documento como de realizar el desarrollo y pruebas del proyecto ha sido un ordenador personal. En el caso de las pruebas de concepto, el ordenador ha sido proporcionado en todo momento por el Grupo de Inteligencia Artificial Aplicada. Obviamente se incluyen todos los elementos necesarios para el uso del mismo: teclado, ratón, monitor, etc.
Cámara SR4000 de MESA Imaging	El sensor utilizado para capturar la información tridimensional. Ha sido proporcionada por el Grupo de Inteligencia Artificial Aplicada. Se puede encontrar información sobre las características del dispositivo en el Anexo I de este documento. Conjunto con la cámara se incluye un trípode y todos los elementos necesarios para su alimentación eléctrica y conexión con el ordenador.

Tabla 1. Hardware utilizado en el proyecto.

Los elementos software requeridos para desarrollar el proyecto han sido los siguientes:

Elemento	Descripción
Windows 7 Professional de 64 bits	El sistema operativo elegido para desarrollar el proyecto ha sido Windows 7 debido a la disponibilidad de las diferentes herramientas requeridas para este sistema. Proporcionado en todo momento por el Grupo de Inteligencia Artificial Aplicada.
Microsoft Visual Studio 2010	Utilizado para el desarrollo del software propio de este proyecto así como para el análisis de la herramienta PCL y el desarrollo de las pruebas con la misma. Proporcionado en todo momento por el Grupo de Inteligencia Artificial Aplicada.
Microsoft Office 2010 Hogar y Estudiantes	Utilizado para el desarrollo de toda la documentación del proyecto, incluido este mismo documento.
Point Cloud Library	Librería estudiada durante el proyecto. Se obtiene gratuitamente a través de internet.
Meshlab	Software de manipulación tridimensional estudiado durante el proyecto. Se obtiene gratuitamente a través de internet.

Tabla 2. Software utilizado en el proyecto.

1.4 Contenido del documento

En el apartado anterior se han descrito los cinco bloques fundamentales que conforman este proyecto. Por tanto, este documento mantiene una estructura similar asignando un capítulo propio a cada uno de dichos bloques, todo ello acompañado por varios capítulos de apoyo y diversos anexos en los que se describen elementos ajenos a los objetivos del proyecto pero relacionados con el mismo de algún modo y que se consideran de utilidad para obtener una visión más global del proyecto.

Definidos los bloques que componen este trabajo, para dar forma a este documento se decide que esté compuesto por los siguientes capítulos:

- **Capítulo 1.** Este mismo capítulo, que sirve como introducción al documento. En él se realiza una introducción al proyecto, una descripción de los objetivos y se detalla estructuradamente el contenido del documento. Se realiza también una breve descripción del material utilizado para llevar a cabo el proyecto.
- **Capítulo 2.** En este capítulo se estudian los precedentes sobre los que parte este proyecto. Se describen brevemente los trabajos de Daniel Sánchez y José Manuel Maestre que sirven como base al proyecto y orientan las pruebas de concepto realizadas respectivamente. Se corresponde con el primer bloque de trabajo definido en los objetivos del proyecto.
- **Capítulo 3.** En este capítulo se realiza un estudio teórico del estado del arte en el campo del escaneado tridimensional. Se realiza un esbozo de los diferentes sistemas de adquisición de datos tridimensionales, las técnicas de procesamiento de datos tridimensionales más utilizadas y los métodos de modelización de los datos obtenidos. Para finalizar el capítulo se incluyen algunas de las aplicaciones prácticas más importantes en la actualidad del escaneado tridimensional de datos. Se corresponde con el segundo bloque descrito en los objetivos del proyecto en el apartado anterior.
- **Capítulo 4.** En este capítulo se describen todos los elementos software que han tomado partida en este proyecto. Se incluye tanto el software desarrollado específicamente para este proyecto como las herramientas MeshLab (2) y PCL (3). Para cada una de las herramientas analizadas se realiza un estudio detallado sobre sus funciones y uso. Se corresponde con el tercer bloque descrito en los objetivos del proyecto en el apartado anterior.
- **Capítulo 5.** En este capítulo se describen las diferentes pruebas de concepto realizadas para este proyecto. Se hace mención a todas las pruebas importantes realizadas, algunas de ellas en colaboración con Manuel Sillero y José Manuel Maestre orientadas al campo de la antropometría. En dicha descripción se detalla el proceso realizado en la captura y tratamiento de datos con las diversas herramientas analizadas en el proyecto. Se corresponde con el cuarto bloque descrito en los objetivos del proyecto en el apartado anterior.
- **Capítulo 6.** En este capítulo se describen las conclusiones a las que se ha llegado tras finalizar el proyecto. Por un lado, se valorarán las diversas herramientas software utilizadas así como las técnicas que implementan. Por otro lado, se hará una valoración técnica de la cámara utilizada para capturar los datos. Por último, se valorará la aplicación de todo lo anterior en el campo de la antropometría debido a la afinidad con

los trabajos de José Manuel. Se corresponde con el quinto y último bloque de trabajo definido en los objetivos del proyecto.

Para finalizar, los anexos incluidos en el documento son los siguientes:

- **Anexo I. Presupuesto del proyecto.** Se realizará un pequeño estudio del presupuesto necesario para llevar a cabo este proyecto.
- **Anexo II. La cámara SR4000.** En este anexo se describen las características técnicas de la cámara SR4000 de MESA Imaging (4).
- **Anexo III. Introducción a la antropometría.** En este anexo se realiza un breve repaso a lo que es la ciencia antropométrica de cara a comprender las valoraciones que se realizan durante las pruebas de concepto relacionadas con esta ciencia.
- **Anexo IV. Proyecto de escaneo tridimensional real.** En este anexo se describe un ejemplo práctico de escaneo tridimensional llevado a cabo por Alejandro Martín Romo en su proyecto fin de carrera “Levantamiento Mediante técnicas de barrido con láser escáner 3d del puente de Vinuesa” (5). Con este ejemplo comprobaremos las técnicas clásicas de escaneo tridimensional en uno de los aspectos en los que más tiempo lleva utilizándose como es el escaneo de construcciones.
- **Anexo V. Código fuente.** En este anexo se incluyen los diferentes códigos fuente a los que se hace mención en el documento.

CAPÍTULO 2. PRECEDENTES

En este capítulo se describirán trabajos que sirven como base y apoyo al proyecto. En primer lugar se hará un breve resumen del trabajo realizado por Daniel Sánchez del Álamo Benguigui en su proyecto fin de carrera “Time-Of-Flight Camera” (1), en el cual realiza un estudio detallado de la cámara SR4000 utilizada en este proyecto y proporciona una librería mediante la cual traslada el API de control de la cámara utilizada en este proyecto desde su lenguaje nativo C++ a la plataforma .NET de forma que puede ser usado con lenguajes como C#. Esta librería es la que se ha utilizado para controlar la cámara en las pruebas realizadas. En segundo lugar, se realizará una aproximación a los trabajos de José Manuel Maestre relacionados con la tecnología TOF que, como se ha mencionado antes, orientan las pruebas de concepto de este trabajo hacia la antropometría.

CAPÍTULO	2.1 PROYECTO FIN DE CARRERA DE DANIEL SÁNCHEZ DEL
2	ÁLAMO
	2.2 COLABORACIÓN CON JOSE MANUEL MAESTRE

2.1 Proyecto Fin de Carrera de Daniel Sánchez del Álamo

Si bien es cierto que la cámara proporciona una API para el trabajo con la cámara en lenguaje C, Daniel Sánchez del Álamo Benguigui realizó como parte de su proyecto fin de carrera una librería que actúa como envoltura de la funcionalidad del API original de la cámara y que permite su uso en lenguajes manejados .NET. La ventaja de los lenguajes .NET como C# (el utilizado para realizar las pruebas de laboratorio) o Visual Basic, es la facilidad de uso de los mismos, sobre todo si se utiliza el entorno de programación Visual Studio de Microsoft. Es posible que se alcanzara mayor rendimiento utilizando librerías nativas y lenguaje C pero debido a la naturaleza del proyecto en el que se busca la realización de diversas pruebas de concepto con diferentes objetivos de análisis se ha decidido utilizar el lenguaje C# para realizar la programación de algunas de las pruebas y por tanto es necesario utilizar la librería programada por Daniel.

2.1.1 Objetivo principal del proyecto de Daniel

El proyecto fin de carrera de Daniel trata, entre otros temas, la aplicación de la tecnología TOF al ocio. Como parte sustancial del proyecto programó una librería, la cual se describirá más tarde, mediante la cual es posible utilizar la API de la cámara SR4000 en un lenguaje .NET como C#. El objetivo de programar esta librería era posibilitar la implementación de un control de interfaz gestual para un videojuego de conducción como prueba de concepto del uso de la tecnología TOF en el ocio.



Ilustración 3. La tecnología TOF de la cámara SR4000 orientada al ocio.

El resultado de esta prueba de concepto es un videojuego que utilizando el framework de desarrollo para videojuegos de Microsoft XNA (6), es controlado mediante gestos utilizando la

cámara SR4000 como elemento de captura. El sistema se basa en la detección de los movimientos y la posición de los puños del usuario de forma que se puede controlar el giro del vehículo de la misma forma que si sostuviésemos un volante, y la aceleración y el freno alejando y acercando los puños respectivamente.

Esta prueba fue todo un éxito demostrando que la tecnología TOF es aplicable al ocio dado que ofrecía una rápida y precisa respuesta a los movimientos del usuario.

2.1.2 El API de la cámara y la librería SwissRanger

Como se ha comentado anteriormente, la librería programada por Daniel Sánchez del Álamo como parte de su proyecto fin de carrera es una librería de envoltura que permite utilizar el API original proporcionado por la cámara, utilizable mediante lenguaje C, en lenguaje C# del framework .NET de Microsoft.

Esta librería recoge la mayoría de las funciones proporcionadas por el API de la cámara, desde la conexión con la misma a la configuración de los parámetros ajustables y pasando obviamente por la adquisición de datos. Otra de las funciones más importantes del API original y que han sido incluidas en esta librería es la posibilidad de conectarte a una cámara virtual que, utilizando un fichero grabado previamente con una captura real de la cámara y generado a través del mismo API, te permite simular la utilización de la cámara sin utilizar realmente el hardware de forma que puede realizarse una grabación de una escena y trabajar sobre esa misma grabación tantas veces como queramos. Esta funcionalidad sin embargo arrastra un problema dado que bloquea la aplicación cuando se llega al final de la secuencia al intentar obtener nuevos fotogramas y no haber recibido señal de parar la adquisición. Es preciso por tanto a la hora de realizar pruebas con esta cámara virtual controlar que se para la adquisición de fotogramas antes de alcanzar el final de la secuencia.

Es necesario recalcar que para el uso de esta librería es necesaria la instalación del driver original de la cámara.

2.2 Trabajos de José Manuel Maestre

Como ya se comentó en la introducción de este documento este proyecto parte de la colaboración realizada con Manuel Sillero, profesor titular en la Facultad de Ciencias de la Actividad Física y del Deporte (INEF-Madrid) y con José Manuel Maestre Rodríguez, alumno de master impartido por Manuel Sillero. Parte de los trabajos de José Manuel consideran la posibilidad del uso de la tecnología TOF aplicada a la antropometría y a la salud.

José Manuel en su trabajo “RECONSTRUCCIÓN TOPOGRÁFICA HUMANA MEDIANTE TIEMPO DE VUELO (ToF)” (7), hace una primera aproximación al uso de la tecnología TOF utilizando cámaras como la SR4000 utilizada en este proyecto en el ámbito de la antropometría y ciencias de la salud. En dicho estudio propone el uso de esta tecnología como opción diferente del método de escaneado tridimensional que se usa actualmente en ese campo, el escáner laser tradicional. Dicha propuesta está basada en algunas de las principales características de la tecnología TOF incluida en cámaras como la SR4000 como pueden ser la velocidad de captura de datos, muy superior a los escáneres laser tradicionales que utilizan un sistema de barrido, y el pequeño tamaño de la cámara, que permite transportarla de forma fácil y cómoda. Por otro lado, menciona el hecho de que no necesita calibración, lo que unido al hecho ya mencionado de que la cámara es fácilmente transportable, permite que el sistema sea portátil y fácilmente instalable, lo que reduce el tiempo necesario de preparación para su uso. Para finalizar hace mención al hecho de que numerosas compañías están dedicando esfuerzos a la producción de diferentes modelos de cámaras con tecnología TOF y a la implementación de mejoras en las mismas a nivel comercial, lo cual es un indicativo de que es una tecnología viable en muchos campos y con un futuro prometedor.

En general, los trabajos de José Manuel centran su carácter técnico en la obtención de volúmenes corporales con los que estudiar fundamentalmente, a nivel antropométrico y de salud, temas relacionados con la obesidad tanto en adultos como en niños. Este tema está muy en boga últimamente y según comenta José Manuel en su trabajo anteriormente mencionado, parece ser necesario incluir nuevas medidas antropométricas a las habituales en este tipo de estudios. Como comenta José Manuel, diversos estudios muestran la necesidad de combinar diferentes medidas antropométricas con el BMI (de las siglas Body Mass Index o Índice de Masa Corporal) para mejorar la predicción de enfermedades cardiovasculares y morbilidad. No obstante, la escasa información existente en este proceder ha favorecido que el BMI sea el método de clasificación nutricional más utilizado. Por su parte, el avance de la tecnología óptica permite conocer de forma rápida y no invasiva diferentes medidas antropométricas que combinadas con el BMI comienzan a superar las desventajas del uso del BMI de forma aislada.

Este trabajo sirve como punto de referencia ya que gracias la colaboración con José Manuel Maestre y Manuel Sillero a la hora de realizar los casos de prueba que utilizará José Manuel para su trabajo de final de Master, se podrá inferir la viabilidad de la tecnología TOF representada mediante la cámara SR4000 así como las diferentes aplicaciones analizadas en un proyecto con aplicación real.

Sin embargo, este trabajo no se orienta a establecer la viabilidad de los elementos aquí descritos en proyectos antropométricos ya que dicho cometido corresponde al trabajo de José Manuel.

En cuanto a este aspecto, este trabajo se limita a describir las conclusiones a las que haya llegado José Manuel en este sentido.

Por otro lado, es destacable en este punto que los resultados a nivel antropométrico que se obtengan en este estudio pueden no ser determinantes en cuanto a la viabilidad de la tecnología TOF como tal, dado que se cuenta únicamente con un modelo de cámara, con unas características específicas que pueden no ser las adecuadas para este cometido o ser mejoradas por otro modelo actual o futuro. En cualquier caso, el producto obtenido generalmente desde cualquier dispositivo de este carácter es una nube de puntos, por lo que cualquier parte de este estudio no relacionado exclusivamente con el modelo de cámara SR4000 utilizado en este proyecto puede ser aplicable en cualquier otro dispositivo.

CAPÍTULO 3. ESTADO DEL ARTE

En este capítulo se realiza un estudio del estado del arte en cuanto a la tecnología de captura tridimensional. Para ello, se describen someramente las diferentes técnicas de adquisición de datos tridimensionales, el procesamiento de dichos datos y la modelización de los mismos. Finalmente se realiza un repaso de las diferentes aplicaciones que esta tecnología tiene en el mundo real.

CAPÍTULO 3

3.1 CAPTURA DE DATOS TRIDIMENSIONALES

3.2 PROCESAMIENTO DE DATOS TRIDIMENSIONALES

3.3 MODELIZACIÓN DE DATOS TRIDIMENSIONALES

3.4 APLICACIONES DEL ESCANEADO TRIDIMENSIONAL

3.1 Captura de datos tridimensionales

Existen numerosas tecnologías utilizadas para la adquisición de datos tridimensionales tanto de objetos como de escenas en general. A continuación se describen someramente los diferentes métodos existentes así como los sistemas más destacados de cada uno de ellos.

3.1.1 Métodos de contacto

Los métodos de obtención de modelos tridimensionales por contacto se basan en el contacto físico sobre el objeto del cual se quiere obtener el modelo tridimensional. Esto implica que sólo es válido para obtener datos sobre un único objeto y no sobre una escena. Los escáneres 3D examinan el objeto apoyando el elemento de medida (palpador) sobre la superficie del mismo, típicamente una punta de acero duro o zafiro. Una serie de sensores internos permiten determinar la posición espacial del palpador. Un CMM (Máquina de medición por coordenadas) o un brazo de medición son ejemplos de un escáner de contacto. Se usan en su mayoría en control dimensional en procesos de fabricación y pueden conseguir precisiones típicas de 0,01 mm. Su mayor desventaja es que requiere el contacto físico con el objeto para ser escaneado, por lo que el acto de escanear el objeto quizás lo modifique o lo dañe. Este hecho es crítico cuándo se escanean objetos delicados o valiosos tales como los artefactos históricos. La otra desventaja de los CMMs es que son muy lentos en comparación con los otros métodos que se pueden utilizar para escanear. El movimiento físico del brazo donde se monta el escáner puede ser muy lento y el CMMs más rápido puede sólo operar en unos pocos cientos de hertzios. Por contraste, un sistema óptico semejante al de un sistema de escáner de láser puede operar de 10 a 1000 kilohertzios.

3.1.2 Métodos pasivos sin contacto

Los métodos pasivos sin contacto basan su funcionamiento en la captación de algún tipo de radiación ambiente reflejada por el entorno. Generalmente se utilizan sistemas que captan la luz visible pero pueden utilizarse otros tipos de radiación como la luz infrarroja. Estos métodos por lo general son bastante baratos dado que pueden utilizarse cámaras digitales convencionales.

Los métodos más comunes son:

- **Sistemas estereoscópicos.** Estos sistemas generalmente utilizan dos cámaras separadas y enfocando la misma escena. La reconstrucción del modelo tridimensional del entorno que enfocan se realiza comparando las diferencias entre las imágenes tomadas por cada cámara de forma que es posible determinar la distancia a cada punto de la imagen. Estos sistemas están basados en la visión humana.
- **Sistemas fotométricos estéreo.** Estos sistemas generalmente utilizan una sola cámara que toma imágenes del objeto bajo diferentes condiciones lumínicas de forma que es capaz de obtener un modelo tridimensional calculando la orientación de la superficie de cada píxel.
- **Sistemas de silueta.** Estos sistemas se basan en el uso de una sola cámara que toma imágenes alrededor del objeto que se quiere modelar bajo condiciones en las que exista un alto contraste entre el objeto y el fondo. De esta forma, se realiza una conjunción de todas las imágenes tomadas generando una aproximación de la superficie exterior del

objeto. El problema de estos sistemas es que son incapaces de detectar concavidades en los objetos.

3.1.3 Métodos activos sin contacto

Los métodos de adquisición activos sin contacto se basan en la emisión de algún tipo de radiación o luz de forma que obtienen datos de un objeto o del entorno detectado el retorno de la luz sobre el objeto o la cantidad de radiación que pasa a través de él. Se utiliza por ejemplo luz, ultrasonidos o rayos-x.

Los métodos más utilizados son los siguientes:

- Time of Flight (Tiempo de vuelo).** Un escáner 3D de tiempo de vuelo determina la distancia a la escena cronometrando el tiempo del viaje de ida y vuelta de un pulso de luz. Un diodo láser emite un pulso de luz y se cronometra el tiempo que pasa hasta que la luz reflejada es vista por un detector. Como la velocidad de la luz C es conocida, el tiempo del viaje de ida y vuelta determina la distancia del viaje de la luz, que es dos veces la distancia entre el escáner y la superficie. Si T es el tiempo del viaje completo, entonces la distancia es igual a $(C * T)/2$. Claramente la certeza de un escáner láser de tiempo de vuelo 3D depende de la precisión con la que se puede medir el tiempo T : 3,3 picosegundos (aprox.) es el tiempo requerido para que la luz viaje 1 milímetro. Se utilizan láseres visibles (verdes) o invisibles (infrarrojo cercano). El distanciómetro láser sólo mide la distancia de un punto en su dirección de la escena. Para llevar a cabo la medida completa, el escáner va variando la dirección del distanciómetro tras cada medida, bien moviendo el distanciómetro o deflectando el haz mediante un sistema óptico. Este último método se usa comúnmente porque los pequeños elementos que lo componen pueden ser girados mucho más rápido y con una precisión mayor. Los escáneres láser de tiempo de vuelo típicos pueden medir la distancia de entre 10.000 y 100.000 puntos cada segundo. Más adelante en esta misma memoria se describirá una cámara basada en esta tecnología que ha sido utilizada para realizar las pruebas oportunas para este proyecto.
- Triangulación.** El escáner láser de triangulación 3D es también un escáner activo que usa la luz del láser para examinar el entorno. El haz de luz láser incide en el objeto y se usa una cámara para buscar la ubicación del punto del láser. Dependiendo de la distancia a la que el láser golpee una superficie, el punto del láser aparece en lugares diferentes en el sensor de la cámara. Esta técnica se llama triangulación porque el punto de láser, la cámara y el emisor del láser forman un triángulo. La longitud de un lado del triángulo definido por la cámara y el emisor del láser es conocida. El ángulo del vértice del emisor de láser se sabe también. El ángulo del vértice de la cámara (paralaje) puede ser determinado mirando la ubicación del punto del láser en la cámara. Estos tres valores permiten determinar el resto de las dimensiones

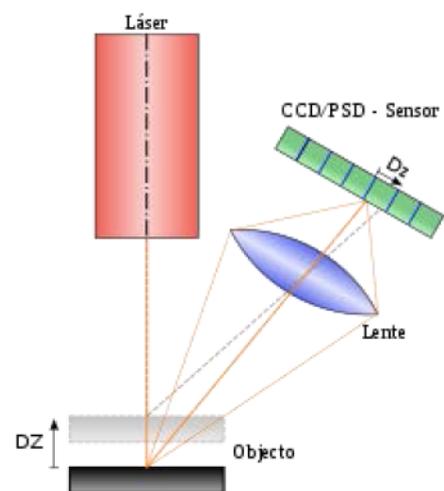


Ilustración 4. Método de triangulación.

del triángulo, y por tanto, la posición de cada punto en el espacio. La precisión de este sistema de medida puede ser muy elevada (milésimas de milímetro), pero depende del ángulo del vértice opuesto al escáner (cuanto más se aparte de 90° más baja es la precisión), lo que limita el tamaño de la escena a analizar. Dado que ese ángulo depende fuertemente de la distancia entre el emisor láser y la cámara, el aumentar el alcance supone incrementar mucho el tamaño del equipo de medida. En la práctica, el alcance máximo de estos escáneres se limita a 20-30 cm. En la mayoría de los casos en lugar de un punto de medida se proyecta una línea que barre la superficie del objeto para acelerar el proceso de adquisición. El Consejo Nacional de Investigación de Canadá fue de los primeros institutos en desarrollar la tecnología en la que se basa el escaneo por triangulación en 1978.

- **Diferencia de fase.** Este tercer tipo de escáner mide la diferencia de fase entre la luz emitida y la recibida, y utiliza dicha medida para estimar la distancia al objeto. El haz láser emitido por este tipo de escáner es continuo y de potencia modulada. El rango y la precisión de este tipo de escáner es intermedio, situándose como una solución entre el largo alcance de los dispositivos de tiempo de vuelo y la alta precisión de los escáneres por triangulación. Su alcance ronda los 200 m en condiciones de poco ruido (baja iluminación ambiente), y su error característico ronda los 2 mm por cada 25 m. En algunos modelos el alcance está limitado precisamente por su modo de funcionamiento, ya que al modular el haz con una frecuencia constante, existe ambigüedad en la medida de la distancia proporcional a la longitud de onda de la modulación utilizada. La precisión de la medida también depende de la frecuencia utilizada, pero de manera inversa a como lo hace el alcance, por lo cual estos conceptos son complementarios, y se debe encontrar un punto de compromiso entre ambos, o bien utilizar dos frecuencias distintas (multi-frequency-ranging). La velocidad de adquisición es muy alta, consiguiendo los modelos actuales velocidades de escaneo que oscilan entre los 100.000 y 1 millón de puntos por segundo, en función de la precisión requerida.
- **Holografía conoscópica.** Es una técnica interferométrica por la que un haz reflejado en una superficie atraviesa un cristal birrefringente, esto es, un cristal que posee dos índices de refracción, uno ordinario y fijo y otro extraordinario que es función del ángulo de incidencia del rayo en la superficie del cristal. Como resultado de atravesar el cristal obtienen dos rayos paralelos que se hacen interferir utilizando para ello una lente cilíndrica. Esta interferencia es capturada por el sensor de una cámara convencional obteniendo un patrón de franjas. La frecuencia de esta interferencia determina la distancia del objeto en el que se proyectó el haz. Esta técnica permite la medición de orificios en su configuración colineal, alcanzando precisiones mejores que una micra. La ventaja de esta técnica es que permite utilizar luz no coherente, esto quiere decir que la fuente de iluminación no tiene por qué ser un láser, la única condición es que sea monocromática. Las aplicaciones de esta técnica son muy variadas, desde la ingeniería inversa hasta la inspección de defectos superficiales en la industria del acero a altas temperaturas. Esta técnica fue descubierta por Gabriel Sirat y Demetri Psaltis en el año 1985 en *“Conoscopic Holography. Applications of Holography”* (8).
- **Luz estructurada.** Los escáneres 3D de luz estructurada proyectan un patrón de luz en el objeto y analizan la deformación del patrón producida por la geometría de la escena. El modelo puede ser unidimensional o de dos dimensiones. Un ejemplo de un modelo

unidimensional es una línea. La línea se proyecta sobre el objeto que se analiza con un proyector de LCD o un láser. Una cámara, desviada levemente del proyector de modelo, mira la forma de la línea y usa una técnica semejante a la triangulación para calcular la distancia de cada punto en la línea. En el caso del modelo de una sola línea, la línea se barre a través del campo del panorama para reunir información de distancia una tira a la vez. Un ejemplo de un modelo bidimensional es una cuadrícula o un modelo de líneas. Una cámara se usa para registrar la deformación del modelo y un algoritmo bastante complejo se usa para calcular la distancia en cada punto en el modelo. Una razón para la complejidad es la ambigüedad. Considere una serie de rayas verticales paralelas de láser que barren horizontalmente a través de un blanco. En el caso más sencillo, uno podría analizar una imagen y asumir que la secuencia izquierda-derecha de rayas refleja la sucesión de los láseres en la serie, así de esta manera la raya de extremo izquierdo de la imagen sea el primer láser, el próximo es el segundo láser, etcétera. En objetivos no triviales que contienen cambio de patrón, hoyos, oclusiones, y de la profundidad, sin embargo, esta secuencia se descompone como rayas que a veces se esconden o pueden aparecer incluso con el orden cambiado, teniendo como resultado la ambigüedad de raya de láser. Este problema particular fue resuelto recientemente por una tecnología de ruptura llamada Multistripe Laser Triangulation (MLT). El escaneo estructurado de luz todavía es un área muy activa de investigación con muchas investigaciones publicadas cada año. La ventaja de los escáneres 3D de luz estructurada es la velocidad. En vez de escanear un punto a la vez, escanean múltiples puntos o el campo entero del panorama inmediatamente. Esto reduce o elimina el problema de la deformación del movimiento. Algunos sistemas existentes son capaces de escanear objetos en movimiento en tiempo real.

- **Luz modulada.** Escáneres 3D de luz modulada emiten una luz continuamente cambiante en el objeto. Generalmente la fuente de luz simplemente cicla su amplitud en un patrón sinodal. Una cámara detecta la luz reflejada y la cantidad que el patrón de luz cambia para determinar la distancia viajada por la luz.

3.2 Procesamiento de datos tridimensionales

El procesamiento de los datos tridimensionales es el trabajo que se realiza como tal sobre los mismos con el objetivo de obtener un resultado determinado. Muchas de las técnicas de procesado pueden llevarse a cabo independientemente del origen de los datos (obtenidos mediante un dispositivo de captura o generados de cualquier otra forma) y tienen muy diversos objetivos. Desde la mejora de los datos capturados mediante diversos algoritmos o técnicas estadísticas hasta la obtención de información de relevancia de acuerdo al objetivo (obtención de un volumen por ejemplo). A continuación se realiza un breve repaso sobre algunas de las técnicas más importantes de procesamiento de datos tridimensionales y que son utilizadas durante las pruebas de concepto.

3.2.1 Filtrado de datos

El proceso de filtrado consiste en la selección de un subconjunto de los datos el cual se considera que es el que aportará información relevante respecto al objetivo final. De esta forma se deshecha cualquier otro dato superfluo reduciendo el conjunto de datos con el que trabajar.

Existen múltiples formas de realizar el proceso de filtrado, tanto manuales como semi-automáticas o automáticas. Los criterios de filtrado pueden ser muy diversos y dependen en gran medida de la información que pueda proporcionarnos el dispositivo de captura utilizado. De esta forma, se puede realizar filtrado por posición, por color o por veracidad de los datos por mencionar algunos. Otro de los procesos de filtrado más importante es el muestreo cuyo objetivo es la selección de los datos más significativos de la muestra independientemente de si se ha realizado ya un proceso de filtrado previo o no. Un ejemplo de muestreo es el descrito en *“Efficient and Flexible Sampling with Blue Noise Properties of Triangular Meshes”* (9).

El objetivo final del proceso de filtrado es que la cantidad de datos disponible sea la menor posible con la mayor relevancia posible y menor pérdida de información posible. Esto es así dado que la potencia de cálculo de las computadoras no es infinita y obviamente se conseguirán resultados más rápidamente reduciendo el conjunto de datos.

3.2.2 Suavizado de superficies

El proceso de suavizado es muy importante cuando la nube de puntos obtenida es muy ruidosa o cuando está compuesta de múltiples capturas que no han sido alineadas perfectamente. Cuando una nube de puntos está obtenida de esta forma, las superficies de los objetos capturados suelen presentar defectos por lo que conviene procesarlas de forma que se reduzca el ruido de la captura.

3.2.3 Alineado

El proceso de alineado consiste en unificar bajo un mismo sistema de referencia varias capturas. Por norma general, los algoritmos desarrollados para este proceso suelen trabajar identificando puntos correspondientes entre las diferentes capturas y encontrando la transformación necesaria que minimiza la distancia entre el conjunto de puntos correspondientes. El proceso suele repetirse hasta que el error es menor que un determinado umbral. Para aumentar la precisión del alineado, suele utilizarse uno o varios elementos marcadores a la hora de capturar la escena de forma que sean fácilmente diferenciables y puedan establecerse con total seguridad como puntos correspondientes entre las diferentes capturas.

Uno de los algoritmos más utilizados es el ICP (Iterative Closest Point) del cual existen multitud de variantes como la descrita en “*A method for registration of 3-D shapes*” (10). El algoritmo ICP tiene como entrada dos nubes de puntos. Una de ellas es considerada la referencia y se mantiene estática mientras que la otra es transformada para alcanzar el mejor alineamiento con la referencia. Puede incluirse de manera opcional una estimación inicial de la transformación necesaria para alinear las dos nubes de puntos. Por otra parte es necesario incluir una condición de parada para el algoritmo. Hecho esto, la secuencia de ejecución es la que sigue:

1. Para cada punto de la nube de referencia, encontrar el punto más cercano en la nube a transformar.
2. Estimar la combinación de rotación y traslación necesarias utilizando una función de mínimos cuadrados para calcular el error de forma que la alineación entre los puntos correspondientes encontrados en el paso anterior sea la mejor posible.
3. Transformar la nube de puntos.

Este proceso se repite hasta que se cumplen los criterios de parada especificados.

3.2.4 Segmentado

El proceso de segmentado consiste en dividir la nube de puntos en diferentes zonas. Los algoritmos de segmentado son especialmente útiles cuando una nube de puntos está compuesta de varias regiones aisladas, casos en los que realizar este proceso permite romper la nube en sus partes constituyentes que pueden ser procesadas de forma independiente.

El caso más habitual de aplicación práctica de esta técnica es el del reconocimiento de objetos.

3.3 Modelización de los datos tridimensionales

La modelización de los datos tridimensionales es otra técnica de procesamiento pero que debido a su importancia se considera oportuno que tenga su propio apartado. La práctica totalidad de sistemas mencionados anteriormente generan como resultado una nube de puntos del objeto o escena analizados. Si bien puede utilizarse directamente para la medición, análisis y visualización de los elementos escaneados, generalmente el objetivo final es generar un modelo tridimensional a partir de dichas nubes de puntos con el que poder trabajar. Esto consiste en darle una superficie consistente al objeto capturado. A continuación se describen los tres tipos de modelos más utilizados.

3.3.1 Modelos de malla poligonal

En una representación poligonal de una forma, una superficie curva es modelada como muchas pequeñas superficies planas (al igual que una esfera es modelada como una bola de discoteca). El proceso de convertir una nube de puntos en un modelo poligonal 3D se llama reconstrucción. La reconstrucción de modelos poligonales implica encontrar y conectar los puntos adyacentes mediante líneas rectas con el fin de crear una superficie continua. Generalmente se utilizan triángulos como polígonos tal y como puede observarse en la reconstrucción del delfín mostrada en la ilustración del margen.

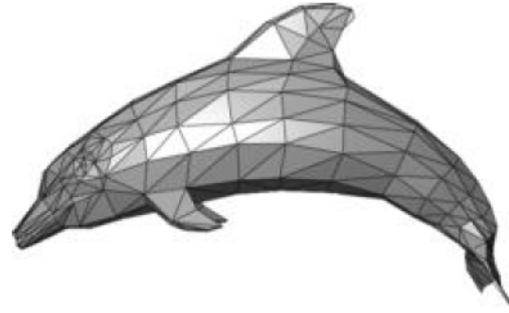


Ilustración 5. Modelo de malla poligonal.

Los modelos poligonales, también llamados modelos de malla, son útiles para la visualización o para algunas aplicaciones CAM, pero son, en general, "pesados" (archivos de datos muy grandes), y son relativamente difíciles de editar en este formato.

Existen muchas aplicaciones, tanto libres como propietarias, destinadas a este fin: MeshLab (2), Cyclone, kubit PointCloud para AutoCAD, JRC 3D Reconstructor, PhotoModeler, ImageModel, PolyWorks, Rapidform, Geomagic, ImageWare, Rhino, etc. En este proyecto se estudió la herramienta MeshLab (2), por lo que más adelante se describirá en profundidad su uso y capacidades.

3.3.2 Modelos de superficies

El siguiente nivel de sofisticación en la modelización implica el uso de un conjunto de pequeñas superficies curvas que unidas entre sí modelan nuestra forma. Estas superficies pueden ser

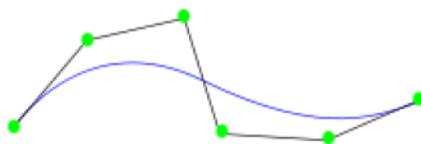


Ilustración 6. Ejemplo de curva NURBS

NURBS, T-Splines u otras representaciones de curvas. Utilizando NURBS, nuestra esfera es una esfera matemática verdadera. En la ilustración 3 se puede observar una curva NURBS (línea azul) que se genera en función de los puntos de control, el vector de nodos y el grado.

Estas superficies tienen la ventaja de ser más ligeras y más fácilmente manipulables cuando se exportan a CAD. Los modelos de superficie son algo más modificables, pero sólo en un sentido escultórico de empujar y tirar para deformar la superficie. Esta representación se presta bien al modelado de formas orgánicas o artísticas.

Algunas aplicaciones sólo ofrecen un diseño manual de las curvas, pero las más avanzadas ofrecen tanto manual como automático. Aplicaciones usadas para este modelado son: Rapidform, Geomagic, Rhino, Maya, T Splines, etc.

3.3.3 Modelos sólidos CAD

Desde el punto de vista de la ingeniería y la fabricación, la representación fundamental de una forma digitalizada es el modelo CAD, totalmente editable. Después de todo, el CAD es el "lenguaje común" de la industria para describir, editar y producir la forma de los bienes de una empresa. En CAD, una esfera está descrita por parámetros que son fácilmente editables mediante el cambio de un valor (por ejemplo, el centro de la esfera o su radio).

Estos modelos CAD no describen simplemente el envoltorio o la forma del objeto, sino que también incorporan la "intención del diseño" (es decir, las características fundamentales y su relación con otras funciones). Un ejemplo de la intención del diseño más allá de la forma por sí sola podrían ser los tornillos de un freno de tambor, que deben ser concéntricos con el agujero en el centro del tambor. Este conocimiento podría guiar la secuencia y el método de creación del modelo CAD: un diseñador con el conocimiento de esta relación, no diseñaría los tornillos referenciados al diámetro exterior, sino que lo haría depender del centro del tambor. Por tanto, un diseñador creando un modelo CAD, incluirá tanto la forma como la finalidad del diseño en el modelo CAD completo.

Distintos enfoques se ofrecen para llegar al modelo CAD. Algunos exportan las superficies NURBS tal cual y dejan que sea el diseñador el que complete el modelo en CAD (por ejemplo, Geomagic, ImageWare, Rhino). Otros utilizan el análisis de los datos para crear un modelo editable basado en las características que se importa en CAD con el árbol de características intacto, produciendo un modelo completo y nativo de CAD, recogiendo tanto la forma como la finalidad del diseño (Geomagic, Rapidform). Mientras que otras aplicaciones de CAD son lo suficientemente robustas como para manipular modelos de un número limitado de puntos o polígonos dentro del entorno CAD (por ejemplo, Catia).

3.4 Aplicaciones del escaneado tridimensional

Existen numerosas aplicaciones para los modelos tridimensionales. Entre ellas, las más importantes son las siguientes:

- **Industria.** Obviamente la industria en general es uno de los grandes usuarios de la modelización tridimensional. El escáner 3D ha encontrado una aplicación insustituible en el control dimensional de fabricación de componentes que requieren tolerancias muy estrictas, como álabes de turbina o los mecanizados de alta precisión. Las piezas se escanean y la nube de puntos se compara con el modelo teórico, permitiendo un control muy minucioso sobre la producción. También se utiliza para "escalar" diseños a partir de modelos creados a mano.
- **Ingeniería inversa.** El objetivo de la ingeniería inversa es la obtención de información o el diseño de un producto accesible al público de forma que pueda investigarse como funciona, de que está hecho y como fue fabricado. Por tanto, la elaboración de un modelo tridimensional de un producto mecánico y de todos los elementos que lo conforman mediante un escáner tridimensional supone una ayuda inestimable en el proceso de ingeniería inversa.
- **Entretenimiento.** Otro de los grandes usuarios de los modelos tridimensionales es el sector del entretenimiento, fundamentalmente en películas y videojuegos. Si bien es cierto que los modelos tridimensionales utilizados podrían generarse directamente por computadora, generalmente escanear un objeto real o incluso esculpirlo y luego escanearlo es mucho más rápido y genera mejores modelos que cualquiera que pudiera generarse por computadora. Actualmente también comienza a verse este tipo de sistemas en el ámbito de la interacción con el usuario, como es el caso de Kinect (11) para Microsoft Xbox.

- **Patrimonio y arquitectura.** Los escáneres tridimensionales son también muy utilizados en el ámbito de la arquitectura en general y del patrimonio histórico, fundamentalmente en esculturas y edificios. En el caso de la arquitectura, se utilizan modelos tridimensionales obtenidos mediante escáner como documentación y para analizar



Ilustración 7. Escaneado del David de Miguel Ángel.

el estado del edificio. En el caso del patrimonio histórico, se utilizan para realizar recreaciones de edificios históricos o de esculturas de forma que contribuyen a la documentación y mantenimiento de dichos elementos así como permiten el estudio y divulgación de los mismos por ejemplo, a través de tours virtuales. Un ejemplo lo podemos ver en el trabajo de Alejandro Martín en el que realiza un modelo virtual del puente romano de la localidad de Vinuesa utilizando un escáner láser topográfico titulado "Levantamiento Mediante técnicas de barrido con láser escáner 3d del puente de Vinuesa" (5).

- **Modelos del terreno.** Cada vez mas se utilizan escáneres tridimensionales para generar modelos tridimensionales del terreno sustituyendo a los sistemas fotogramétricos convencionales. Se utiliza fundamentalmente el laser, dado que este sistema en su forma aerotransportada sufre menos influencia por las condiciones atmosféricas adversas, como cobertura de nubes y lluvia. Como se trata de luz próxima del espectro visible, las interrupciones visuales del pulso son los únicos obstáculos en el proceso. De esa forma, días de poco sol son incluso más propicios para la ejecución de levantamientos láser. Otra ventaja es la rapidez en la captación.

CAPÍTULO 4. SOFTWARE PARA EL PROCESADO TRIDIMENSIONAL

En este capítulo se realiza una descripción en profundidad del software considerado de utilidad para el desarrollo de aplicaciones que trabajen con datos tridimensionales. Comienza con la descripción del software desarrollado específicamente para el proyecto y continúa con las herramientas que se decidió analizar, MeshLab (2) y PCL (3), ambas software libre disponible para cualquier proyecto de desarrollo y con objetivos suficientemente distintos como para ser interesante el análisis de ambas herramientas.

CAPÍTULO 4

4

4.1 SOFTWARE DESARROLLADO

4.2 MESH LAB

4.3 PCL (POINT CLOUD LIBRARY)

4.1 Software desarrollado

En este apartado se describe el software desarrollado específicamente para este proyecto para servir de apoyo al mismo. El objetivo de este software es facilitar el tratamiento de las capturas realizadas con la cámara SR4000 proporcionando una primera opción de filtrado y segmentación así como facilitando la conversión de los datos a los formatos soportados por MeshLab y PCL. Este software es específico para este modelo de cámara en particular.

4.1.1 Análisis previo

Como se ha explicado en el capítulo anterior, se parte del trabajo realizado por Daniel Sánchez en su trabajo fin de carrera que permite la utilización de la cámara TOF SR4000 en el entorno .NET y más concretamente, en lenguaje C#. Realizando un análisis profundo de este trabajo se detecta un problema fundamental que dificulta trabajar con las secuencias obtenidas durante el uso de la cámara.

A pesar de que la librería permite grabar secuencias, su reproducción se basa en emular un dispositivo virtual que simula una grabación en tiempo real. Este sistema presenta dos problemas fundamentales. En primer lugar, solo permite la navegación hacia delante entre los diferentes frames capturados por lo que no resulta práctico a la hora de poder analizar frame a frame en las que muchas veces es de utilidad volver al anterior. Por otra parte, la cámara virtual presenta un gran inconveniente que hace que su uso directo sea complicado. En versiones anteriores del driver, al llegar al final de la captura el dispositivo se bloqueaba dejándolo inutilizable. Esto ha sido solucionado en versiones posteriores pero en estas, la secuencia se repite de forma indefinida hasta que el usuario la para manualmente de forma que es complicado saber cuál es el final de la secuencia. Por estos motivos, se decidió como punto de partida desarrollar un formato de fichero y un programa que permitiese almacenar y visualizar los frames de una secuencia de forma que no resulte necesaria la utilización del driver y del programa propietario de la cámara.

4.1.2 Formatos de almacenamiento utilizados

A continuación se describen los diferentes formatos que se utilizan en las pruebas realizadas.

Formato SRS

Este formato es el que proporciona la propia librería de la cámara para almacenar secuencias y que puede ser reproducido utilizando una cámara virtual que simula una adquisición en tiempo real. Esta funcionalidad ha sido portada en el proyecto realizado por mi compañero. Sin embargo, se desconoce por completo el funcionamiento interno de dicho sistema así como el formato de fichero. Dicho sistema de hecho tiene varios defectos que han llevado a implementar un nuevo formato e incluir en la nueva librería la funcionalidad de almacenaje. Estos problemas son:

- El formato del fichero es desconocido por lo que se hace imprescindible utilizar el software de la cámara para trabajar con los ficheros almacenados.
- El método de apertura de ficheros proporcionado se basa en el uso de una cámara virtual. El principal defecto de este sistema es que produce un bloqueo no recuperable una vez que se visualiza el último frame de la secuencia y se pide uno nuevo.
- No permite navegar hacia atrás al analizar la secuencia de frames.

Formato TSF

Para subsanar los problemas mencionados, se decidió crear un formato para ficheros de secuencias tomadas con la cámara TOF e implementar la funcionalidad necesaria para poder utilizarlo. Este formato almacena todos los datos proporcionados por la cámara TOF para cada frame, de forma que se pueda disponer de toda la información posible para realizar las operaciones que se estimen oportunas. Este sistema almacena frame por frame de la secuencia y podría utilizarse tanto para almacenar secuencias de video como secuencias de fotogramas aislados, de forma que resulta bastante versátil. Dado que su cometido es almacenar la información y no se contempla que su reproducción sea similar a la de un video, no se ha incluido ni información ni funcionalidad necesaria para que las secuencias pudieran ser reproducidas a la velocidad a la que fueron tomadas.

En la siguiente tabla se muestra la definición del formato del archivo, determinando los campos que contiene, su utilidad y el tamaño de los mismos.

<p>CABECERA</p> <ul style="list-style-type: none"> • Número mágico: definido para comprobar si el fichero es válido. 4 bytes. • Número de frames: indica el número de frames contenido en el fichero. 4 bytes.
<p>FRAME 0 ... N</p> <ul style="list-style-type: none"> • Por cada pixel: 144x176 = 25344 pixels <ul style="list-style-type: none"> • Datos 3D: Los datos 3D del pixel. <ul style="list-style-type: none"> ○ X: 4 bytes ○ Y: 4 bytes ○ Z: 4 bytes • Datos amplitud: 2 bytes • Datos confianza: 2 bytes • Datos distancia: 2 bytes

Tabla 3. Formato de fichero TSF

Los ficheros almacenados de esta forma no son legibles utilizando un programa como el bloc de notas ya que los datos están codificados a nivel de byte y no almacenados como caracteres ASCII. Si se hubiera determinado almacenar cada uno de los valores como una serie de caracteres ASCII, el tamaño del fichero se hubiera disparado.

Una vez que tenemos definido el formato de fichero, podemos realizar una estimación del tamaño que podría alcanzar un fichero con este sistema al almacenar una secuencia.

Dado que el formato de fichero está totalmente determinado, se puede calcular el tamaño del fichero de forma trivial utilizando la siguiente fórmula:

$$Tamaño = Cabecera + n^{\circ} Frames(Tamaño Frame)$$

De esta forma, pasamos a calcular el tamaño de cada frame cuyo valor será constante y que se determina mediante la siguiente fórmula:

$$Frame = (Alto \times Ancho) \times (X3D + Y3D + Z3D + Amplitud + Confianza + Distancia) \\ = (144 \times 176) \times (4 + 4 + 4 + 2 + 2 + 2) = 456192 \text{ bytes}$$

Por otro lado, tenemos en cuenta que la cámara es capaz de tomar secuencias a velocidades de hasta 25 frames por segundo. Siendo conservadores, tomamos que la cámara graba una

secuencia a 20 frames por segundo, de lo que obtenemos el tamaño de cada segundo de grabación mediante la siguiente fórmula:

$$\text{Segundo} = \text{Frame} \times \text{FPS} = 456192 \times 20 = 9123840 \text{ bytes}$$

De la fórmula anterior obtenemos que cada segundo de captura almacenado tiene un tamaño de 9123840 bytes, o redondeando, alrededor de 9 megabytes. A priori es un tamaño elevado, superior al tamaño de las secuencias almacenadas utilizando el fichero original SRS de la librería proporcionada por la cámara. Sin embargo, hay que tener en cuenta que el fichero SRS solo almacena los datos capturados y que no incluye aquellos que se obtienen mediante cálculos que realiza el propio driver de la cámara, como ocurre con los datos de coordenadas tridimensionales para cada punto. En cualquier caso, debido al coste actual del almacenamiento, no se ha considerado que esto sea un problema prioritario pero se puede plantear como futura ampliación la posibilidad de comprimir la información de cada uno de los frames contenidos en la secuencia de forma que el tamaño total se reduzca.

Formato ASC

El formato ASC es un formato utilizado por muchos programas de modelado 3D para almacenar y recuperar la nube de puntos correspondiente a una escena u objeto. Es muy simple dado que almacena los datos en formato ASCII en el que cada punto se almacena en una línea con 3 columnas separadas por el carácter “,” que se corresponden con los valores X, Y y Z de cada punto.

Debido a que este formato es utilizado por un gran número de programas y puede ser considerado con un estándar, la librería programada contiene la funcionalidad necesaria para almacenar nubes de puntos en este formato. En la siguiente tabla se puede ver la simpleza de este formato:

PUNTO 0 ... N ValorX,ValorY,ValorZ (Valores en ASCII separados por “,”)

Tabla 4. Formato de fichero ASC

Este formato almacena la nube de puntos de un único frame. Calcular el tamaño que puede tomar este fichero para un frame es complejo a priori debido a que en muchos casos no se almacenan todos los puntos del frame ya que suelen aplicarse filtros que eliminan aquellos puntos que no nos ofrecen información de utilidad. Por otra parte, al almacenarse como caracteres ASCII, cada valor tiene un número de cifras diferente por lo que es imposible calcular la longitud real del fichero. En cualquier caso, la siguiente fórmula puede resultar una aproximación al tamaño máximo tomando en cuenta que se almacena la nube completa y que cada valor X, Y o Z utiliza 10 caracteres:

$$\begin{aligned} \text{Nube} &= \text{Alto} \times \text{Ancho} \times (X + 1 + Y + 1 + Z) = 144 \times 176 \times (10 + 1 + 10 + 1 + 10) \\ &= 811008 \text{ bytes} \end{aligned}$$

De la fórmula anterior, podemos determinar que el tamaño de una nube completa se aproxima a 800 kilobytes.

Formato PCD

Este formato es propio de la librería PCL que se ha mencionado anteriormente y que se estudiará en el próximo capítulo. Por ello, se ha desarrollado una funcionalidad en la librería programada que nos proporciona las nubes de puntos en este formato. En la siguiente tabla se muestra la estructura de este tipo de fichero de forma simplificada y será descrito por completo en su capítulo correspondiente:

<p>CABECERA</p> <ul style="list-style-type: none"> • Versión: especifica la versión del fichero. • Campos: especifica los nombres de los campos que contienen las dimensiones y otros valores. • Tamaño: especifica el tamaño de cada dimensión en bytes. • Tipo: especifica el tipo de dato de cada dimensión. • Cuenta: especifica cuantos datos tiene cada dimensión para un punto dado. • Ancho: número de puntos almacenados con datos desorganizados. • Alto: número de filas. • Punto de vista: punto desde el que se ha tomado la imagen. • Puntos: especifica el número de puntos totales. • Datos: especifica el formato en el que se almacenan los datos.
<p>PUNTO 0 ... N</p> <ul style="list-style-type: none"> • ValorX ValorY ValorZ (Valores en ASCII separados por un espacio)

Tabla 5. Formato de fichero PCD

Como se ha mencionado, es una descripción sencilla del formato que será explicada en profundidad más adelante debido a que permite muchas variantes. En este caso, se ha optado por la representación que se ha utilizado para transformar las nubes de puntos obtenidas por la cámara a este formato.

Al igual que en el caso anterior, calcular el tamaño total del fichero es complicado por los mismos motivos que para el formato ASC. En cualquier caso, debido a su semejanza se puede estimar que el tamaño será muy similar.

4.1.3 Herramientas de filtrado

Una vez que hemos obtenido una secuencia es bastante útil filtrar los puntos que nos interesan y eliminar aquellos que no aportan información a nuestros propósitos. Para ello, se ha realizado una serie de filtros de forma que podamos obtener una nube de puntos que solo contenga aquellos que nos interesan. A continuación se describirán los diferentes filtros programados así como se mostrará un ejemplo de la utilización de los mismos.

Para los ejemplos, se utiliza como imagen de muestra una de las capturadas durante las pruebas de concepto, mostrando el valor de amplitud tomado por la cámara. Se utilizan los datos de amplitud debido a que es lo más parecido a una imagen en escala de grises de la escena tomada:



Ilustración 8. Imagen previa al filtrado

Filtro de confianza

Como se ha comentado anteriormente, uno de los valores que se obtienen al tomar una imagen con la cámara es el valor de confianza para cada píxel. Dicho valor nos indica cuán confiable es la medida tomada por la cámara para ese píxel. Este valor de 16 bits, representa una mayor confianza cuanto mayor es el valor almacenado. Por tanto, realizar este filtro es trivial, dado que se establece un límite en la confiabilidad y se descartan todos los píxeles cuyo valor de confiabilidad sea inferior al límite marcado. En la siguiente ilustración podemos ver la imagen de ejemplo filtrada con una confiabilidad del 80%.

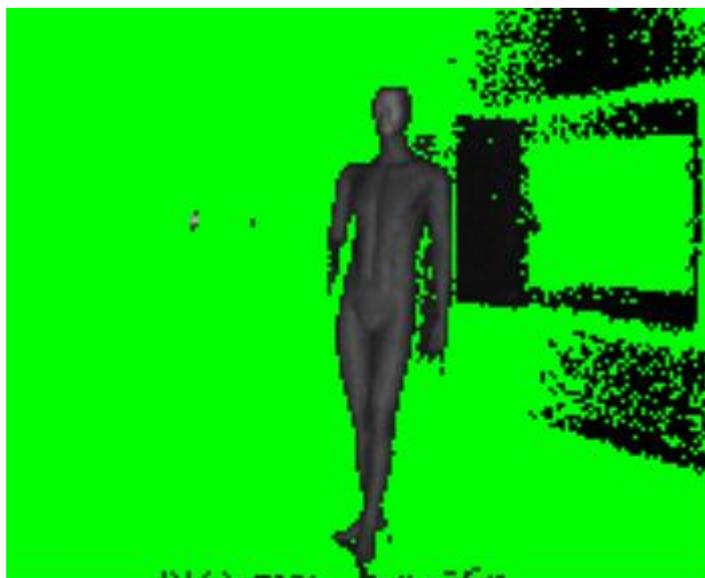


Ilustración 9. Imagen de ejemplo filtrando la confiabilidad al 80%

Como se puede observar, se eliminan (pasan a color azul) todos los puntos cuya confiabilidad está por debajo del 80%. Se puede ver como se eliminan fundamentalmente aquellos puntos lejanos y aquellos otros que recogen una superficie demasiado brillante (como el suelo).

Filtro de distancia

Con este filtro se eliminan aquellos puntos que están más lejos o más cerca de lo que queremos analizar. Para ello, este filtro examina el valor de distancia de cada píxel y descarta todos aquellos que se encuentran fuera del rango requerido. Al igual que en el caso anterior y como se explicó anteriormente, este valor es ofrecido por la cámara mediante 16 bits que representan la distancia desde cada píxel al objetivo siendo a mayor valor, mayor distancia. En la siguiente imagen se muestra la imagen de ejemplo filtrando los puntos de forma que solo se conserven aquellos que están en la distancia del maniquí que es el objeto que nos interesa.

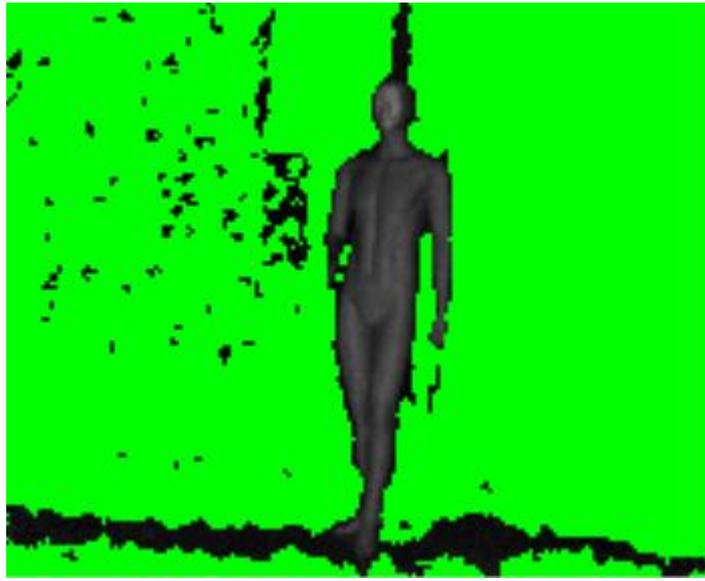


Ilustración 10. Imagen de ejemplo filtrando la distancia

Como se puede observar, se eliminan aquellos puntos que están por delante o por detrás del maniquí. Existen algunos puntos que a priori deberían ser eliminados pero que no lo son debido a que el valor de distancia puede no ser confiable y expresar una distancia errónea. Se puede ver comparando esta imagen con la anterior que la mayor parte de los puntos que debería haber sido eliminados si lo han sido tras el filtrado de confianza.

Filtro rectangular

Este filtro es bastante simple y a la vez muy útil cuando queremos segmentar un objeto en particular y descartar el resto de la escena. Para ello, se determina un rectángulo en el que todos los píxeles interiores son válidos y todos los externos son descartados. En la siguiente imagen podemos ver el resultado de aplicar este filtro.

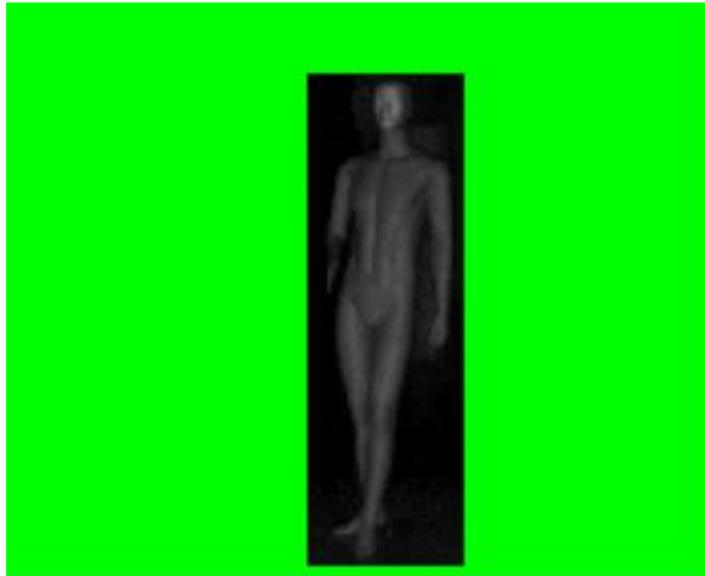


Ilustración 11. Imagen de ejemplo con filtro rectangular

Podemos ver la simpleza y a la vez la utilidad de este filtro. Se ha descartado la escena dejando solamente el rectángulo que contiene al maniquí.

Filtro de puntos aislados

La función de este filtro es eliminar todos aquellos puntos que no tienen vecinos a una distancia determinada. Esto quiere decir que, una vez aplicados otros filtros y eliminados ciertos puntos, se analiza la imagen y elimina aquellos puntos que se han quedado aislados del resto de la imagen. La aplicación de este filtro es configurable dado que se puede definir el número de vecinos mínimo para que el píxel sea válido y la distancia radial al píxel para calcular el número de vecinos. En el siguiente ejemplo podemos ver como actuaría este filtro.

Podemos ver una imagen de ejemplo en la que dos bloques de 4 píxeles han quedado aislados del resto de la imagen.

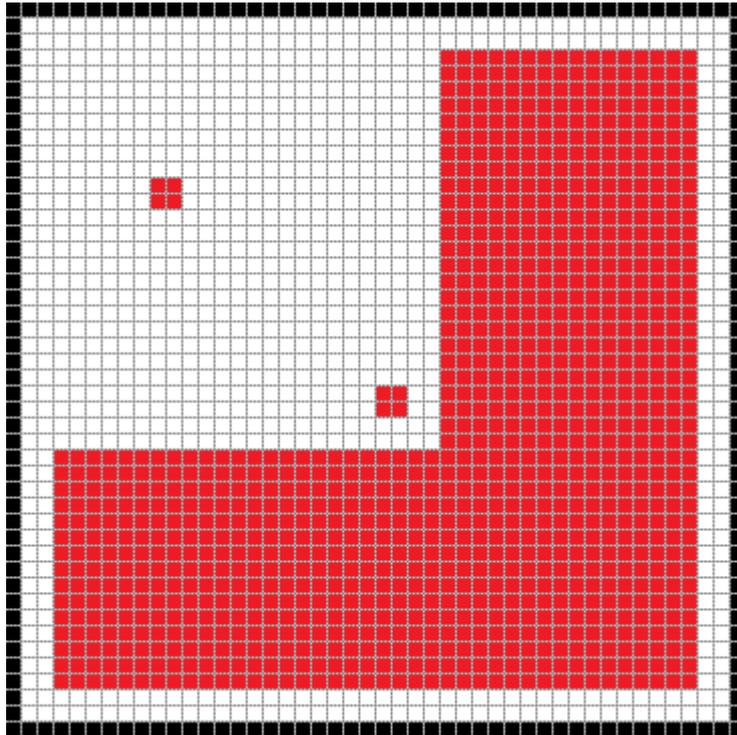


Ilustración 12. Imagen de ejemplo previa al filtro de puntos aislados

A continuación, la misma imagen si realizamos un filtro de puntos aislados en el que decidimos eliminar aquellos puntos que tienen menos de 4 vecinos a una distancia de 4 píxeles de sí mismos.

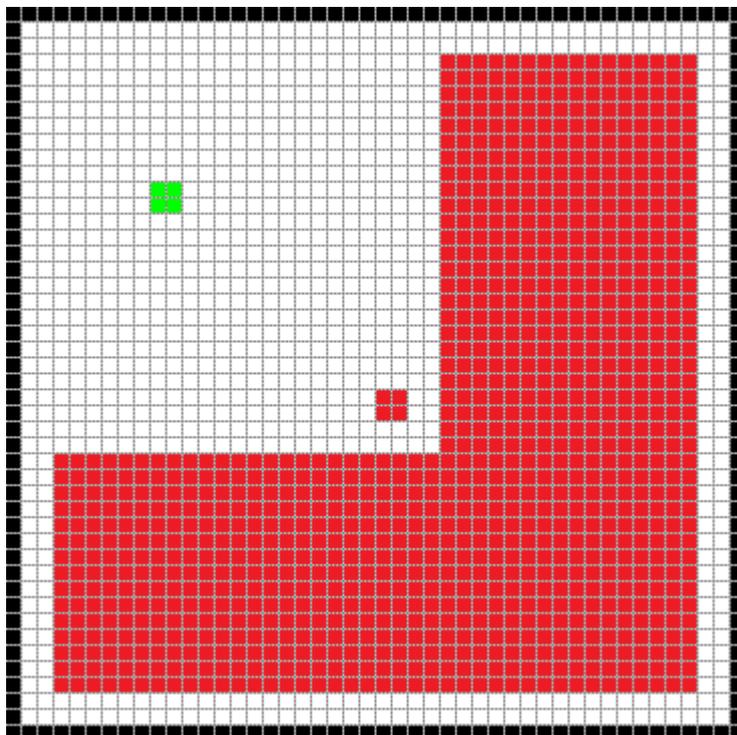


Ilustración 13. Imagen de ejemplo posterior al filtro de puntos aislados

Como podemos observar, el bloque más alejado de lo que se conserva de la imagen ha sido eliminado mientras que el bloque más cercano aún cumple las condiciones requeridas para que se mantenga. El funcionamiento de este filtro es muy similar a la operación morfológica matemática de erosión que se aplica generalmente en imágenes bidimensionales. Podemos encontrar información bastante completa sobre operadores morfológicos en el artículo Morfología matemática de Wikipedia (12). Su objetivo es eliminar aquellos puntos que, debido a su aislamiento, no nos aportan información de relevancia.

Filtro de saturación

Este filtro se utiliza para aquellos puntos que, una vez obtenida la secuencia, la cámara ha sido incapaz de determinar su posición real debido a la saturación. Para aplicar este filtro, basta con eliminar todos aquellos puntos cuyas coordenadas X, Y y Z sean (0,0,0), valor con el que la cámara marca los píxeles saturados.

Otros posibles filtros no implementados

Si bien se han implementado algunos filtros que nos permitiesen obtener datos útiles para las siguientes pruebas del proyecto, hay muchos otros filtros que se han considerado pero que no han sido implementados debido a que no era parte de los objetivos del proyecto. A continuación se describen someramente:

- **Filtro de selección libre.** Este filtro tendría como objetivo seleccionar aquellas zonas de la imagen que queremos analizar, al igual que el filtro rectangular. Sin embargo, su uso se asemejaría a la herramienta de selección de forma libre del programa Paint de Microsoft, de forma que se pudiera seleccionar una forma más compleja que un rectángulo.
- **Filtro de puntos aislados tridimensionalmente.** Este filtro, de características similares al filtro de píxeles aislados implementado, tendría como objetivo descartar aquellos píxeles aislados respecto a los demás en la nube de puntos basándose en la distancia de cada pixel a los demás. Esto podría eliminar aquellos puntos que, debido a errores en la medición, resultan en coordenadas totalmente fuera de lugar.

Aplicación conjunta de los filtros

Realizar un filtrado de una escena con un único filtro es útil pero si se combinan más de uno se obtienen los mejores resultados. A continuación vamos a comprobar el resultado de aplicar todos los filtros implementados para seleccionar únicamente el maniquí y aquellos puntos del mismo cuya confiabilidad sea alta.

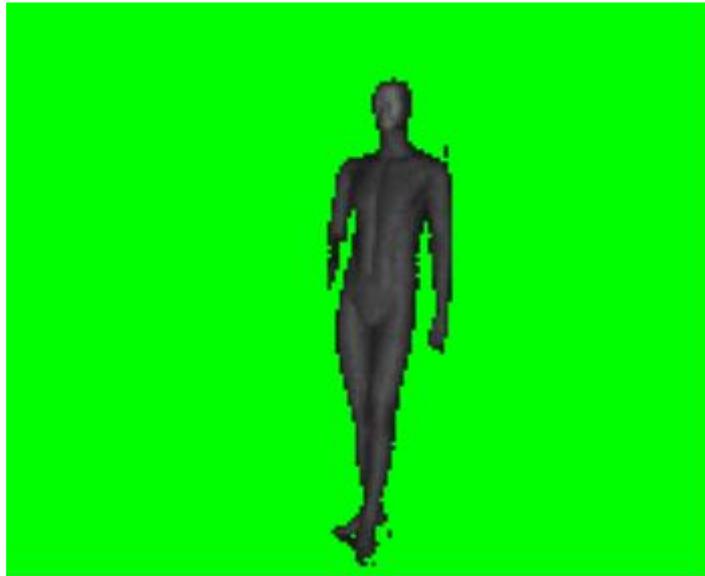


Ilustración 14. Utilización conjunta de los filtros

Como podemos observar, prácticamente solo permanecen aquellos píxeles que pertenecen al maniquí, de forma que podemos analizar el objeto que nos interesa de la escena.

4.2 MeshLab

MeshLab es un programa de procesamiento de modelos tridimensionales muy conocido en el mundillo del desarrollo de modelos y manejo de datos tridimensionales. El software es desarrollado por el centro de investigación ISTI – CNR (Istituto di Scienza e Tecnologie dell'Informazione) que es un instituto italiano de investigación de la Universidad de Pisa. Su objetivo principal es posibilitar el procesamiento de los modelos tridimensionales no estructurados que suelen provenir del escaneado tridimensional. Así mismo, provee de una serie de herramientas para editar, inspeccionar, renderizar y convertir estos modelos.



Ilustración 15. Logotipo de MeshLab.

4.2.1 Descripción

MeshLab incluye un gran número de herramientas de todo tipo que nos permiten trabajar con las nubes de puntos o modelos que hayamos obtenido mediante el escaneado tridimensional. Contiene herramientas tanto manuales como automáticas siendo estas últimas muy poderosas. MeshLab es introducido en “*MeshLab: an Open-Source Mesh Processing Tool*” (13).

En primer lugar, MeshLab nos permite visualizar y movernos con total libertad sobre una nube de puntos o sobre un modelo, todo en un entorno tridimensional. Gracias a esto podemos realizar un análisis exhaustivo de los datos procesados. La potencia de este software nos permite visualizar conjuntos de datos de gran tamaño sin problemas.

En segundo lugar, entre las herramientas manuales que nos proporciona el software, se nos permite la selección de puntos o caras de la nube o modelo de forma que, una vez realizada la selección oportuna, podemos realizar diferentes tareas como pintar o eliminar los elementos seleccionados. Por otro lado, MeshLab nos permite realizar mediciones entre dos puntos del modelo o nube. El resultado de esta medición vendrá dado por el sistema de coordenadas utilizado así como de la escala aplicada. De esta forma, si obtenemos una nube de puntos con la cámara utilizada en este proyecto, como nos ofrece los resultados en un sistema de coordenadas tomado en metros, podemos realizar una comparación entre la precisión de la cámara y la realidad.

Por otra parte, MeshLab contiene multitud de herramientas automáticas de gran potencia que nos permiten interactuar y manipular las nubes de puntos o modelos. A continuación se mencionan y explican de forma breve algunas de las herramientas que se han considerado más útiles:

- **Filtros de limpieza.** El software nos permite aplicar diferentes filtros que nos eliminen de forma automática aquellos elementos que considera ajenos al modelo o irrelevantes para el mismo. Entre otros, MeshLab es capaz de detectar y eliminar de forma automática puntos duplicados, componentes aislados no relevantes para el modelo o incluso rellenar agujeros que considera inexistentes en el modelo real.
- **Filtros de mallado.** Con estos filtros, es capaz de generar o corregir modelos mallados, fundamentalmente obtenidos a través de nubes de puntos. Entre otras tareas, MeshLab

es capaz de generar superficies a partir de una nube de puntos, simplificar modelos, dividir superficies o suavizar modelos. Esto último es de gran utilidad dado que elimina el ruido que se pueda producir al escanear superficies.

- **Colorización.** MeshLab nos permite realizar algunas tareas de colorización de modelos que resultan de utilidad para analizar modelos o nubes de puntos.
- **Integración tridimensional.** El software permite alinear varias nubes de puntos de forma manual o mediante el algoritmo ICP que se explicará más adelante. Gracias a esto, es capaz de unificar el espacio de referencia de varias nubes de puntos obtenidas de la misma escena.

Por último, una característica de MeshLab muy importante para cualquier desarrollador es que el software puede ser utilizado como programa completo o bien como librería. Esto es importantísimo dado que permite realizar pruebas rápidas para cualquier proyecto mediante la típica interfaz gráfica y, más tarde, utilizar toda la potencia del programa MeshLab haciendo uso del mismo como librería en nuestro software.

4.2.2 Plataformas soportadas

Otra de las grandes virtudes de este programa es que puede ser utilizado en una gran variedad de plataformas diferentes que incluyen tanto dispositivos de escritorio como dispositivos móviles. MeshLab no solo funciona en plataformas Windows, Linux y Mac OS, que suelen ser habituales en la mayoría de software de código abierto sino que es capaz de funcionar en dispositivos móviles con sistemas Android o iOS. Esto último provee de gran valor a la herramienta, fundamentalmente si es utilizada como librería dado que permite su uso en proyectos software orientados a dispositivos móviles si bien hay que mencionar que la funcionalidad en estos sistemas es por ahora reducida.

4.2.3 Licencia

El programa MeshLab es gratuito, de código abierto y está sujeto a la licencia GPL (GNU General Public License) (14). El único requisito que se pide desde la web es que en el caso de utilizar MeshLab para un proyecto comercial o de investigación se haga una pequeña mención del mismo en el foro de la herramienta.

Gracias a estas condiciones se considera que puede ser de gran utilidad en cualquier proyecto relacionado con el tratamiento de nubes de puntos, modelos tridimensionales o similares.

4.2.4 Instalación

La herramienta MeshLab puede descargarse accediendo a la dirección web <http://meshlab.sourceforge.net/>. En dicha página podrán encontrarse los diferentes enlaces de descarga para cada una de las diferentes plataformas y sistemas operativos soportados. También es posible encontrar diversa documentación y acceso al código fuente del programa.

Centrándonos en el sistema operativo Windows, sistema con el que se ha trabajado durante el proyecto, instalar el programa MeshLab es muy sencillo e intuitivo. Una vez que hemos realizado la descarga del instalador del programa, basta con ejecutar y seguir el típico asistente de instalación en el que nos solicitarán nuestra conformidad con la licencia del programa así como la carpeta en la que deseemos realizar la instalación, todo ello en un proceso bastante trivial.

En el caso del sistema operativo Linux, la descarga nos facilita el código fuente que deberemos compilar en nuestra propia máquina si bien es cierto que existen paquetes precompilados para algunas distribuciones.

Por otra parte, al no disponer de un sistema Mac OS no ha sido posible realizar pruebas sobre este sistema.

4.2.5 Funcionamiento

Para utilizar el programa MeshLab basta con ejecutar el mismo a través del acceso directo que se genera durante la instalación.

Una vez el programa se encuentra en ejecución observaremos el entorno general del mismo, con una barra de herramientas superior en la que podemos acceder a la multitud de filtros, herramientas de renderizado, vistas, etc., así como una barra de acceso rápido a las herramientas de edición. Por otro lado, en la zona central se encuentra el espacio de trabajo en el que aparecerán los modelos tridimensionales, nubes de puntos, etc. El punto de vista en la zona de trabajo puede ser manipulado de diversas formas. Entre otras, se puede realizar zoom o mover el punto de vista de la cámara, posibilidades básicas que nos permiten cualquier movimiento para visualizar el modelo.

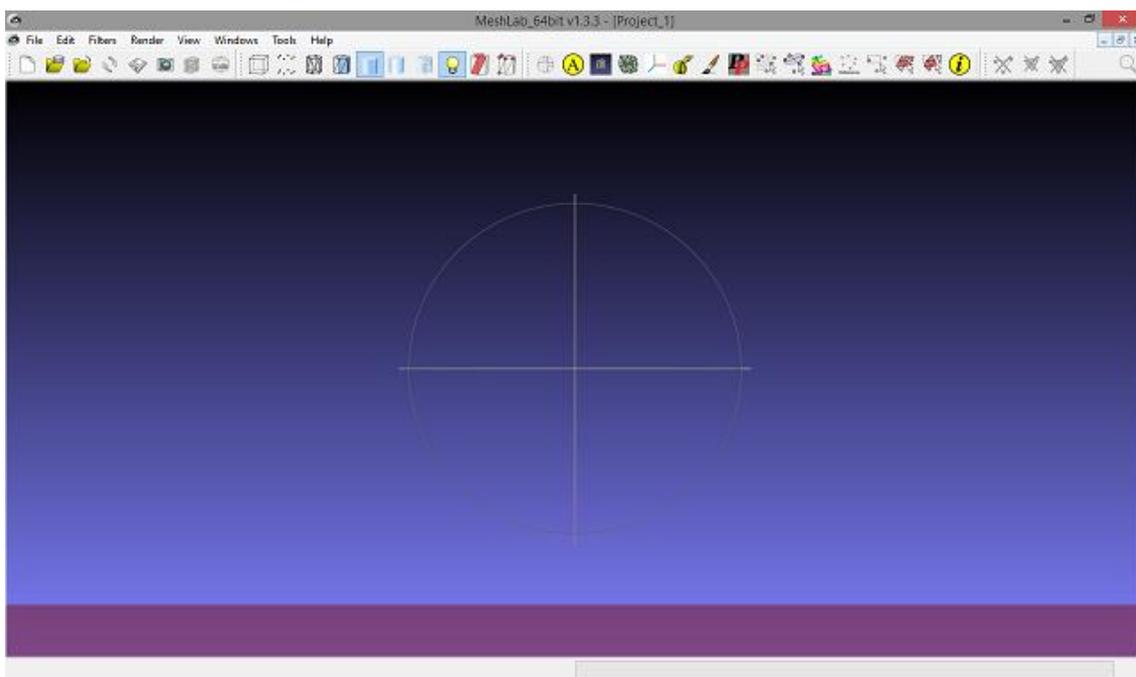


Ilustración 16. Visión general de MeshLab.

Respecto al modelo de trabajo en MeshLab, la unidad básica de trabajo es el proyecto. Se considera que un proyecto es una combinación de uno o más modelos o mesh. A parte de las operaciones básicas como crear, guardar, abrir o cerrar un proyecto, en MeshLab se pueden realizar algunas tareas adicionales. Todas ellas se pueden encontrar en la pestaña File de la barra de menús.

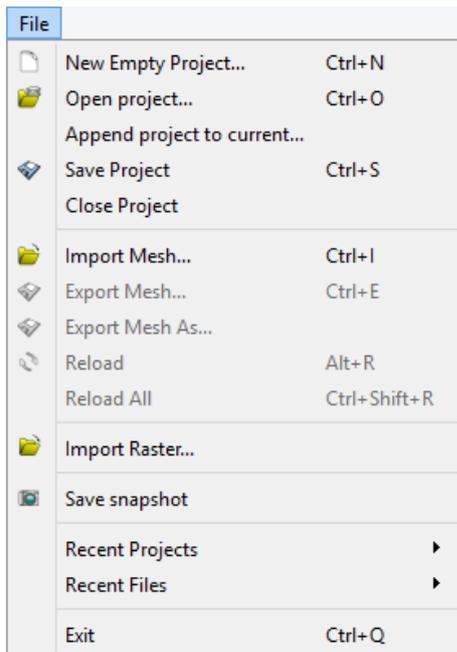


Ilustración 17. Pestaña File en MeshLab.

En primer lugar, MeshLab permite combinar dos proyectos de forma que los modelos de cada proyecto pasen a formar un único proyecto que los contenga a todos. Por otra parte, como es lógico, MeshLab permite importar y exportar los diferentes modelos que pudiera contener el proyecto. Otra de las opciones que permite realizar es la de importar un Raster, una representación matricial de una imagen. Para finalizar, permite tomar una instantánea de lo mostrado en el espacio de trabajo.

El tratamiento que hace MeshLab de cada uno de los modelos incluidos en el proyecto es similar al tratamiento que se realiza en los programas de edición de imágenes digitales como pueden ser Photoshop o Gimp. En estos programas se puede trabajar con lo que se llama capas, que son diferentes secciones de la imagen independientes entre sí de tal forma que puede

aplicarse una operación únicamente a una de las capas en lugar de a toda la imagen. En la siguiente captura podemos observar como MeshLab trata de forma independiente a cada uno de los modelos importados.

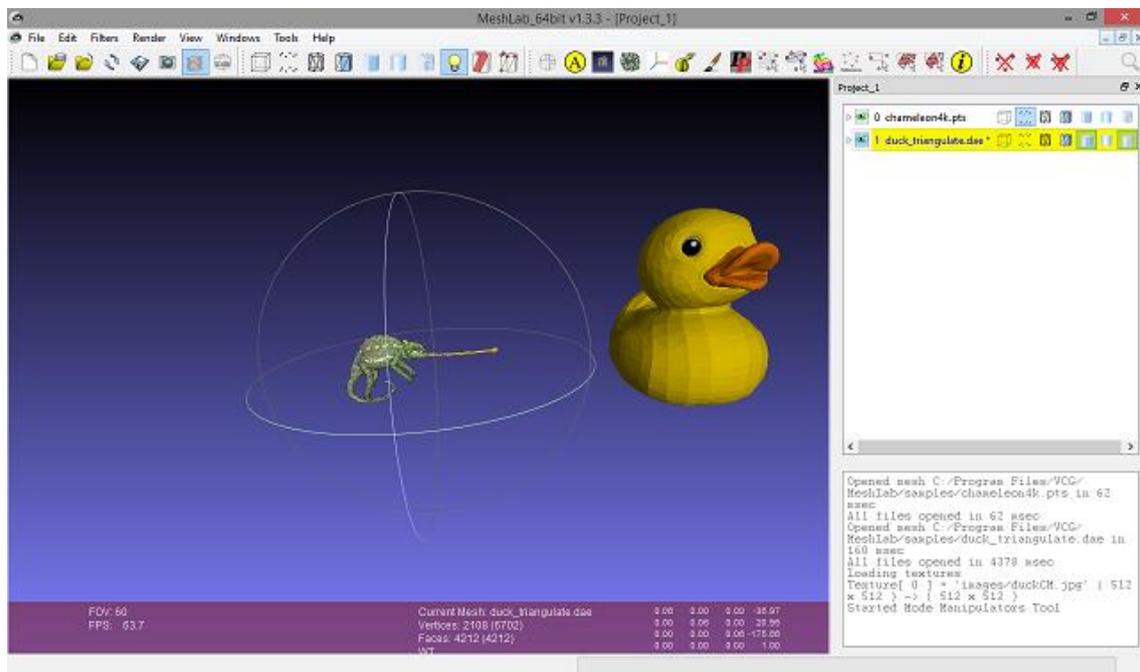


Ilustración 18. Vista de capas.

En la ilustración anterior podemos observar dos modelos de ejemplo importados, un camaleón y un patito. MeshLab permite trabajar con ellos de forma independiente como si se tratara de capas.

MeshLab soporta una gran cantidad de formatos lo que permite que sea pueda utilizar ficheros que provengan de muy diversas fuentes. A continuación se especifican los formatos que soporta:

- **Importación:** PLY, STL, OFF, OBJ, 3DS, COLLADA, PTX, V3D, PTS, APTS, XYZ, GTS, TRI, ASC, X3D, X3DV, VRML, ALN.
- **Exportación:** PLY, STL, OFF, OBJ, 3DS, COLLADA, VRML, DXF, GTS, U3D, IDTF, X3D.

En la pestaña Edit podemos encontrar las operaciones básicas de edición que nos permite realizar el programa. La mayoría de las opciones de este menú son bastante auto descriptivas. Por mencionar alguna de las más importantes, la herramienta “Align” permite alinear dos o más modelos en función de los puntos de coincidencia entre los mismos. Por otro lado, permite la selección de puntos, vértices y caras de diferentes formas así como la realización de operaciones como rotación, traslación y escalado de los modelos. Otra opción importante y de gran utilidad en el caso del estudio de la precisión de aparatos de escaneado es la herramienta de medida. Esta es una herramienta simple que permite medir la distancia entre dos puntos seleccionados. En casos como en el de la cámara utilizada en este proyecto en el cual el sistema de coordenadas ofrece una correspondencia 1:1 entre la medida real y la coordenada obtenida del escaneado nos permite comparar la medida real en el objeto con la obtenida en el modelo sin realizar ningún tipo de operación matemática que iguale los sistemas de coordenadas.

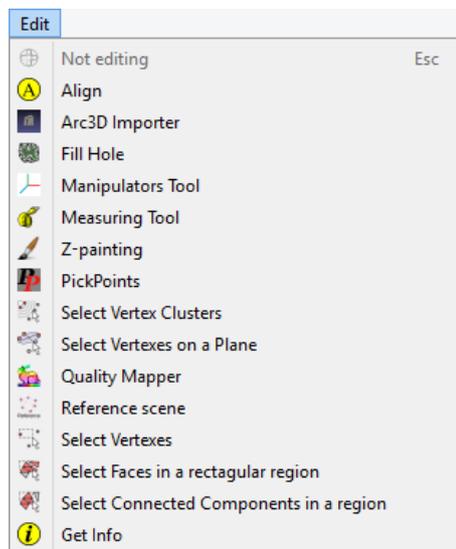


Ilustración 19. Pestaña Edit en MeshLab.

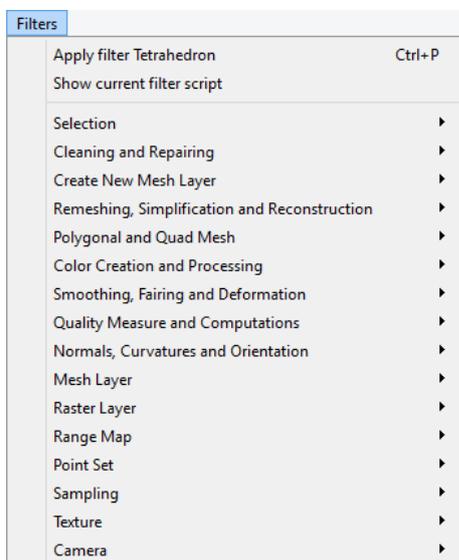


Ilustración 20. Pestaña Filters en MeshLab.

La pestaña Filters de la barra de menús puede considerarse como la más importante del programa MeshLab ya que es la que contiene las utilidades más potentes e importantes de la aplicación. Contiene una inmensa cantidad de opciones correspondientes a diferentes algoritmos, incluidas diferentes versiones de un mismo algoritmo, para realizar diferentes operaciones sobre los modelos tridimensionales. En total existen más de 100 herramientas que permiten realizar muy diversas acciones. Dada la cantidad, a continuación se realiza un repaso de los diferentes grupos de herramientas explicando la utilidad de los algoritmos contenidos en cada uno de ellos pero sin entrar en detalle en el funcionamiento de los mismos para no extender demasiado. En cualquier caso, manteniendo brevemente el ratón sobre cualquiera de

las herramientas aparecerá un cuadro explicativo de la funcionalidad de dicha herramienta así como el fichero DLL (en el caso de Windows) que contiene dicha funcionalidad.

- **Selection.** Las herramientas contenidas en este grupo permiten realizar selección de elementos como vértices, aristas y caras de diversas formas.

- **Cleaning and Repairing.** Las herramientas de este grupo permiten realizar diferentes labores de limpieza y reparación de los modelos tridimensionales. Entre otras, permiten eliminar elementos duplicados (como pueden ser vértices con las mismas coordenadas, caras solapadas, etc.), eliminar elementos aislados o reducir el número de vértices fusionando vértices próximos entre sí.
- **Create New Mesh Layer.** En este grupo se contienen las herramientas para crear diferentes modelos tridimensionales de objetos geométricos como tetraedros, cajas o conos, así como elementos generados mediante diversos algoritmos.
- **Remeshing, Simplification and Reconstruction.** Este grupo de herramientas permite realizar diferentes labores de optimización sobre los modelos así como la generación de los mismos partiendo de una nube de puntos.
- **Polygonal and Quad Mesh.** Diferentes herramientas mayormente para convertir modelos formados por caras triangulares en modelos formados por caras cuadradas o rectangulares.
- **Color Creation and Processing.** Este grupo contiene diversas herramientas cuya función es la creación y procesado del color en los diferentes elementos del modelo teniendo en cuenta en algunos de los algoritmos la incidencia de la luz, la oclusión ambiental, etc.
- **Smoothing, Fairing and Deformation.** Este grupo de herramientas contiene diferentes opciones para realizar operaciones con los modelos tridimensionales cuyo objetivo fundamental es la transformación de la superficie. Por ejemplo, posee herramientas para suavizar dicha superficie o deformarla.
- **Quality Measure and Computations.** Las herramientas contenidas en este grupo están relacionadas con la computación y medida de diferentes elementos de calidad.
- **Normals, Curvatures and Orientation.** En este grupo de herramientas podemos encontrar diversas opciones para generar los vectores normales tanto de vértices como de caras así como la posibilidad de generar las curvaturas y realizar operaciones sobre la orientación, posición y escala del modelo tridimensional.
- **Mesh Layer.** Este grupo de herramientas permite realizar diversas operaciones sobre las capas de modelos de las que se habló a la hora de describir el tratamiento por capas que puede realizar MeshLab. Entre otras, permite eliminar, fusionar o dividir capas en función de diversos parámetros así como mover entre capas diversos elementos del modelo.
- **Raster Layer.** Este es un grupo de operaciones simple para trabajar con raster. Permite eliminar la capa actual de raster o todas ellas, renombrarlas o configurar la cámara para trabajar con este sistema.
- **Range Map.** Este grupo de herramientas solo contiene dos. La primera de ellas es utilizada para alinear dos modelos que representen parte del mismo objeto o escena mientras que la segunda es un algoritmo de selección de caras basado en los vectores normales de cada cara y la dirección del punto de vista.
- **Point Set.** Este grupo permite realizar operaciones con el conjunto de puntos tales como computar los vectores normales de los mismos o realizar proyecciones de los mismos.
- **Sampling.** Este grupo de herramientas contiene diversos algoritmos cuya función generalmente es la toma de puntos de muestra de un modelo tridimensional.

- **Texture.** Este grupo permite realizar diversas operaciones con las texturas de un modelo tridimensional en caso de que las posea, así como asignar texturas a los mismos.
- **Camera.** Con las herramientas contenidas en este grupo se pueden realizar operaciones diversas con la cámara de visualización del programa.

Como ya se comentó anteriormente, al mantener el puntero del ratón sobre cualquiera de las herramientas de los grupos anteriormente descritos aparece un cuadro explicativo de la misma. En dicho cuadro aparece una breve descripción del uso que se le da a dicha herramienta así como una referencia al algoritmo en el que está basada. De igual forma, aparece la librería DLL que contiene dicha funcionalidad, lo que puede ser especialmente útil a la hora de realizar un desarrollo personalizado utilizando solo las herramientas requeridas.

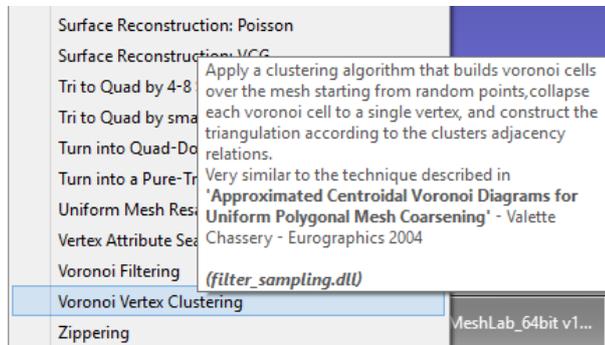


Ilustración 21. Ejemplo de ayuda en MeshLab.

La pestaña Render de la barra de menús contiene diversas opciones referidas a la visualización de los modelos tridimensionales en el espacio de trabajo, concretamente a las opciones de renderizado. Estas opciones incluyen la modificación de la luz, aparición de sombras, gestión del color o modo de renderizado. Respecto a las sombras, hay que destacar que MeshLab permite utilizar una gran cantidad de algoritmos diferentes en su generación de forma que podemos elegir el que más nos convenga en función de la relación velocidad/calidad o simplemente el que nos parezca más realista.

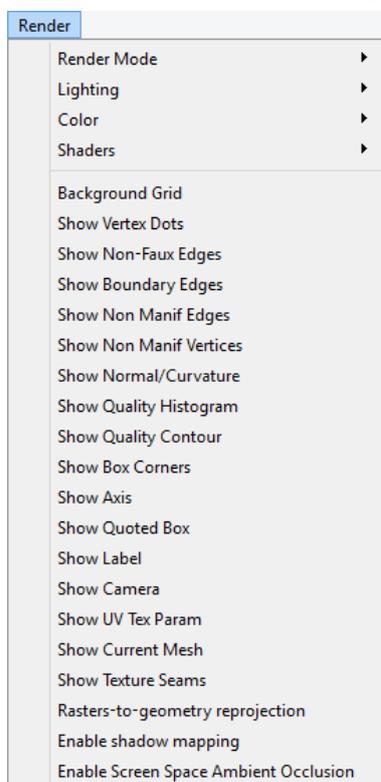


Ilustración 22. Pestaña Render en MeshLab.

Por otra parte, permite activar o desactivar diversas características cuya función es mostrar u ocultar diversos elementos de los modelos tridimensionales como pueden ser el propio eje de coordenadas, la rejilla de fondo, etc. En general, permite mostrar elementos del modelo generalmente matemáticos que no pertenecen al objeto u escena física pero que se generan a partir de él. Destacar que alguna de estas opciones como *“Enable Screen Space Ambient Occlusion”* requieren gran procesamiento por lo que en equipos poco potentes pueden producir una reducción de la velocidad del programa.

Para finalizar, las pestañas View, Windows, Tools y Help son las comunes a este tipo de programas por lo que no ofrecen ninguna funcionalidad destacable y no se considera necesario realizar ninguna explicación al respecto.

4.3 PCL (Point Cloud Library)

PCL o Point Cloud Library es, como su propio nombre indica, es un proyecto a gran escala cuya finalidad es el procesamiento de imágenes y nubes de puntos tanto bidimensionales como tridimensionales. El conjunto de PCL contiene un gran número de algoritmos entre los que se incluyen filtros, estimación de características,

reconstrucción de superficies, registro, ajustes de modelos o segmentación. Podemos encontrar todo tipo de información sobre esta librería en su página oficial (<http://www.pointclouds.org>).



Ilustración 23. Logotipo de PCL.

Este proyecto pertenece a la fundación Open Perception (<http://www.openperception.org>) cuyo objetivo es el desarrollo de software de percepción 3D de forma profesional y utilizando las últimas novedades de dicho campo. Relacionado con este objetivo tiene el propósito de formar usuarios y diseminar información relevante para simplificar el uso y desarrollo de algoritmos y herramientas complejas. Por otro lado, esta agrupación persigue involucrar a diversos socios comerciales e investigadores que soporten y desarrollen una solución abierta y estándar para el procesado de nubes de puntos.

4.3.1 Descripción

La librería PCL es un compendio de múltiples algoritmos y funciones que permiten trabajar con imágenes y nubes de puntos de 2 o 3 dimensiones. Estructuralmente, la librería PCL está dividida en una serie de pequeñas librerías que pueden ser compiladas y utilizadas de forma independiente. Esta es una de las grandes ventajas de PCL dado que de esta forma se consiguen dos objetivos. Por un lado, se simplifica el desarrollo debido a que pequeñas partes de código son más fáciles de mantener que grandes librerías. Por otro lado, se permite el uso de la librería en plataformas con especificaciones reducidas en capacidad o potencia computación. En la siguiente figura se muestra una ejemplificación de la división de PCL mediante un grafo de dependencias.

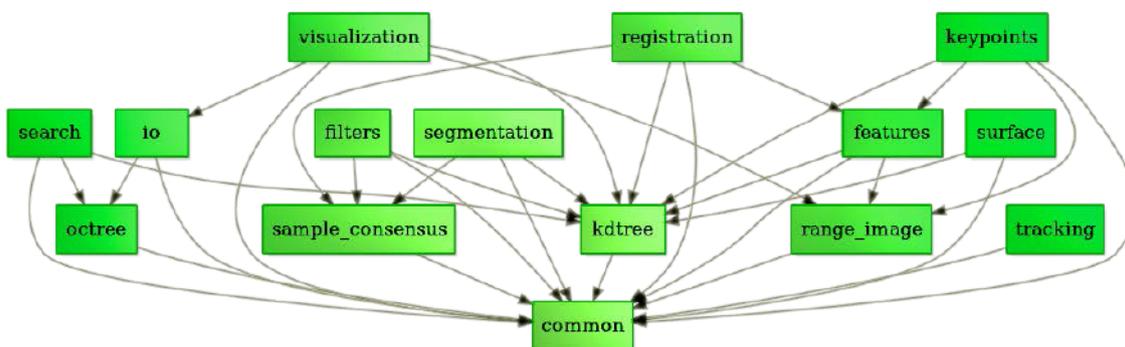


Ilustración 24. Estructura de PCL.

Hay que mencionar que el grafo anterior no es completo pero puede observarse como se encuentra estructurada la librería de forma que no es necesario utilizar todos los módulos para realizar una funcionalidad. En la documentación de la librería se encuentran todas las

dependencias entre los diferentes módulos de forma que es fácil saber que necesitamos para llevar a cabo nuestro proyecto sin necesidad de incluir funcionalidades superfluas.

En el siguiente listado se describen los módulos más importantes de la librería. Estos módulos son los que contienen los algoritmos y herramientas realmente útiles para realizar trabajos con nubes de puntos mientras que se ha obviado describir los módulos que actúan como soporte básico a los demás, como puede ser el módulo `pcl_common` que aparece en el grafo anterior.

Los módulos más importantes de la librería son los siguientes:

- **Filters.** El módulo `pcl_filters` contiene diversos mecanismos para eliminar elementos aislados y ruido en aplicaciones de filtrado de nubes de puntos tridimensionales. En la ilustración del lateral se puede observar un ejemplo de filtrado en el que la imagen de la izquierda presenta diversos puntos producto de errores de medición, cosa que complica el análisis de la nube de puntos. Entre otros, se pueden usar procedimientos estadísticos para eliminar estos puntos erróneos.
- **Features.** El módulo `pcl_features` contiene estructuras de datos y mecanismos para estimar características (conocidas por 3D features) en una nube de puntos. Una característica es una representación en un punto concreto o en una posición en el espacio que describe geoméricamente patrones basados en la información disponible alrededor del punto. El espacio de datos seleccionado alrededor del punto suele llamarse el vecindario k (k -neighborhood en inglés). Algunos ejemplos de características son la normal de un punto o la curvatura de una superficie.
- **Keypoints.** El módulo `pcl_keypoints` contiene implementaciones de dos algoritmos de detección de puntos clave (keypoints en inglés) en nubes de puntos. Los puntos clave o puntos de interés son puntos de una imagen o nube de puntos que son estables, distintivos y pueden ser identificados usando algún criterio determinado de detección. Estos puntos en conjunción con otros parámetros pueden ser usados para formar una representación compacta de los datos originales.
- **Registration.** El módulo `pcl_registration` implementa una serie de algoritmos utilizados para combinar diferentes fuentes de datos en un modelo global consistente utilizando una técnica que en inglés es llamada registration. La idea de este método es identificar los puntos correspondientes entre las diferentes fuentes de datos y encontrar una transformación que minimice la distancia entre dichos puntos. El proceso se repite hasta que el error de alineación entre los puntos baja de un determinado umbral.

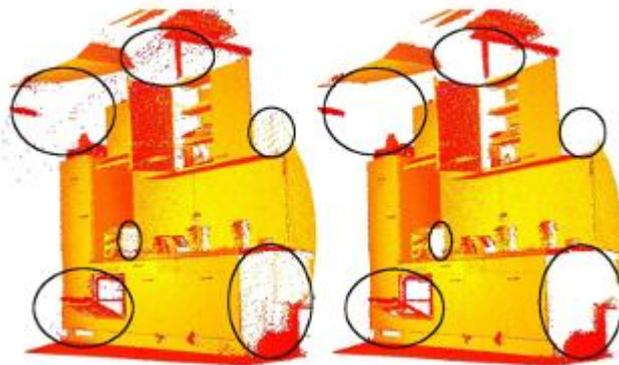


Ilustración 25. Ejemplo de filtrado con PCL.

- **Kdtree.** El módulo `pcl_kdtree` proporciona la estructura conocida como kd-tree o árbol-kd que permite la búsqueda rápida de vecinos. Un árbol-kd es una estructura de partición del espacio que almacena una serie de puntos k-dimensionales en una estructura de árbol que permite una búsqueda eficiente por distancia y vecinos más próximos. La búsqueda del vecino más próximo es una de las operaciones más importantes cuando se trabaja con nubes de puntos y puede usarse para encontrar correspondencias entre grupos de puntos o descriptores de características, o para definir el vecindario local alrededor de un punto o puntos.

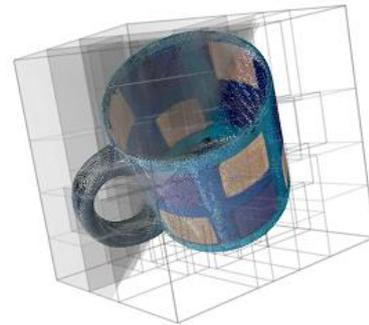


Ilustración 26. Ejemplo de árbol-kd.

- **Octree.** El módulo `pcl_octree` provee métodos para crear una estructura de árbol jerarquizada desde una nube de puntos. Esto permite particionamiento espacial, reducción del muestreo y operaciones de búsqueda en el conjunto de datos. Cada nodo de octree puede tener ocho hijos o ninguno. El nodo raíz describe una envoltura en forma de cubo que encapsula todos los puntos. Cada nivel del árbol subdivide el espacio en un factor de 2 por lo que se incrementa la resolución de voxels o envolturas. La implementación de octree en PCL permite realizar diferentes tipos de búsqueda del vecino más cercano de forma eficiente.
- **Segmentation.** El módulo `pcl_segmentation` contiene algoritmos para segmentar una nube de puntos en diferentes clusters. Estos algoritmos se aplican principalmente en el procesamiento de nubes compuestas por varias regiones aisladas espacialmente. En estos casos, el segmentado se usa para dividir la nube en sus partes constituyentes que pueden ser procesadas de forma independiente.

- **Sample_consensus.** El módulo `pcl_sample_consensus` contiene métodos como RANSAC y modelos como planos y cilindros. Pueden ser combinados libremente para detectar modelos específicos y sus parámetros en las nubes de puntos. El ajuste de planos suele ser utilizado en la tarea de detectar superficies interiores comunes como muros, suelos o mesas. Otros modelos pueden ser utilizados para detectar objetos como tazas haciendo uso de un cilindro. En la ilustración del lateral podemos ver un ejemplo en el que diferentes objetos han sido detectados (marcados con diferente color) utilizando estas técnicas.

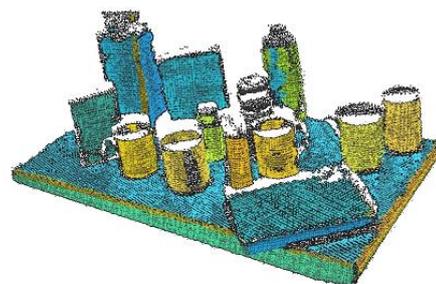


Ilustración 27. Ejemplo de `Sample_consensus` en PCL.

- **Surface.** El módulo `pcl_surface` se encarga de la reconstrucción de las superficies originales desde escaneados tridimensionales, es decir, la creación del modelo mado tridimensional. Es importante utilizar suavizado y filtrado si la nube de puntos contiene ruido o si está compuesta de varios escaneos que no estén correctamente alineados.
- **Recognition.** El módulo `pcl_recognition` contiene diferentes algoritmos utilizados en aplicaciones de reconocimiento de objetos.

- **Io.** El módulo `pcl_io` contiene clases y funciones para leer y escribir ficheros en formato PCD así como capturar nubes de puntos desde una gran variedad de dispositivos. Acepta fundamentalmente dispositivos compatibles con el estándar OpenNI pero algunos otros como la cámara SR4000 utilizada en este proyecto ha sido probada con PCL.
- **Visualization.** El módulo `pcl_visualization` tiene el propósito de ofrecer la posibilidad de realizar rápidamente un prototipo y visualizar los resultados de los algoritmos que operen sobre la nube de puntos tridimensional. Sin embargo, esto no impide realizar una librería de visualización propia dependiendo de los objetivos de la aplicación a desarrollar.

Como se puede ver dadas las características incluidas en la librería, es un proyecto de gran magnitud y no solo es producto de un grupo de programadores interesados en la materia sino que es respaldado por grandes compañías. Este proyecto es desarrollado y utilizado por una multitud de organizaciones y profesionales de la informática y la investigación entre los que cabe destacar la compañía desarrolladora de hardware gráfico Nvidia, el fabricante de automóviles TOYOTA o la desarrolladora de procesadores Intel, así como un gran número de universidades entre las que se encuentra la Universidad Carlos III de Madrid en la cual se han desarrollado varios proyectos haciendo uso de la librería. Muchas de estas compañías no solo contribuyen en su desarrollo sino que lo financian por lo que parece tener un futuro prometedor.

4.3.2 Plataformas soportadas

La librería PCL es multiplataforma y ha sido compilada y desplegada satisfactoriamente en las plataformas de escritorio Linux, Windows y MacOS así como en las plataformas móviles Android e iOS. PCL ha sido desarrollado dividiendo la librería en pequeños componentes que pueden ser compilados de forma autónoma, lo que permite utilizar la librería en plataformas con capacidades reducidas.

4.3.3 Licencia

La librería PCL es liberada bajo los términos de la licencia BSD (15) y es un software de código abierto. Es totalmente gratuita tanto para uso comercial como para uso en investigación.

Destacar en este caso el hecho de que es un gran proyecto de código abierto y solicitan continuamente la colaboración tanto de organismos y empresas como de particulares para su desarrollo por lo que cualquier interesado con conocimientos en el campo de la programación o del escaneado y procesamiento tridimensional puede colaborar en el proyecto.

4.3.4 Instalación

Para poder utilizar PCL es necesario descargarse el paquete de instalación de la librería o el código fuente para compilarlo nosotros mismos. Ambas opciones se encuentran disponibles en la página web <http://www.pointclouds.org/downloads/>. En dicha página podemos encontrar los binarios precompilados de instalación para algunas distribuciones Linux, Windows y Mac OS.

En el caso de Windows, existen dos opciones diferentes de instalación. Por un lado, se encuentra disponible el paquete de instalación completo que contiene PCL y todas sus dependencias excepto la librería Qt, que es utilizada para crear interfaces gráficas. Por otro lado, podemos elegir cada una de las dependencias a instalar y luego compilar el conjunto de módulos de PCL que queramos utilizar. Todas las opciones disponibles se encuentran en la página web

<http://www.pointclouds.org/downloads/windows.html> . En general, se recomienda utilizar el paquete completo que se encuentra disponible para Microsoft Visual Studio 2008 o 2010 para sistemas operativos tanto de 32 bits como de 64. Para esta prueba se opta por utilizar el paquete completo además de la librería Qt.

Para empezar, instalamos la librería Qt. Debemos descargar la versión correspondiente a Visual Studio 2008 o 2010, según el entorno que utilicemos. Para otros entornos será necesario compilar las fuentes o dirigirse a la web del proyecto Qt por si distribuyeran otro tipo de binarios precompilados. En cualquier caso, una vez descargado el instalador apropiado, basta con ejecutarlo, elegir el destino de la instalación y proceder a instalarlo. Una vez terminado, dispondremos de la librería y de diversas herramientas para crear interfaces de usuario. Hecho esto, pasamos a instalar la librería PCL que, como en el caso anterior, se encuentra disponible para Visual Studio 2008 o 2010 aunque de igual forma podemos compilar las fuentes para cualquier otro tipo de entorno. Una vez que hemos realizado la descarga, ejecutamos el instalador y nos aparecerá el típico asistente. En este asistente, además de las típicas opciones en las que se acepta la licencia o se elige el destino de la instalación, aparece la opción de incluir la librería en el path del sistema. En caso de que queramos añadirlo al path, nos dará la oportunidad de ejecutar desde línea de comandos cualquier herramienta de la librería desde cualquier ubicación del sistema. El activar esta opción o no ya queda a decisión de cada cual. En este caso se decide aplicarlo pues resulta más cómodo. Más adelante, otra elección importante es la de que componentes y dependencias de la librería se quieren instalar. Como ya se mencionó, PCL está desarrollado de forma modular por lo que es posible instalar solamente aquellos módulos que se van a utilizar. En este caso, se decide instalar toda la librería y sus dependencias. Durante el proceso aparecerá el asistente de instalación de la librería OpenNI, que no presenta ninguna dificultad añadida. Una vez finalizado, disponemos de la librería y de las diversas herramientas que contiene.

En el caso de Linux, los binarios precompilados se encuentran disponibles para las distribuciones Ubuntu, Debian, Fedora, Gentoo y Arch Linux. En estos casos basta con utilizar el sistema de gestión de software propio de cada distribución estando, en algunos casos, incluidos en los repositorios oficiales de la distribución. En cualquier caso, las instrucciones de instalación concretas para cada sistema se encuentran en <http://www.pointclouds.org/downloads/linux.html> .

Para el caso de Mac OS el proceso de instalación consiste en utilizar los paquetes de instalación precompilados que contienen las dependencias de PCL y posteriormente instalar la librería como tal. Este proceso no ha sido probado al no disponer de un sistema Mac OS para realizar pruebas pero en cualquier caso, dadas las instrucciones proporcionadas en la página web, parece un proceso trivial. El proceso y los ficheros pueden encontrarse en la página web <http://www.pointclouds.org/downloads/macosx.html> .

4.3.5 Funcionamiento

PCL es exclusivamente una librería y como tal, debe formar parte de otro programa para explotar sus capacidades. Existen varias formas de crear un proyecto de configurar un proyecto de desarrollo de software con la librería PCL en función del sistema operativo y el entorno de

desarrollo escogido. En este caso, se ha decidido utilizar como sistema operativo Windows y como entorno de desarrollo Visual Studio 2010.

Para esta plataforma escogida, existe un tutorial en la documentación de PCL disponible en la página web http://www.pointclouds.org/documentation/tutorials/using_pcl_pcl_config.php. Este tutorial es muy genérico y hace uso de herramientas externas para generar una solución o proyecto utilizable por Visual Studio 2010 pero puede servir para iniciarse en PCL. En cualquier caso, a continuación se describe un método para configurar un entorno de desarrollo para una aplicación que utilice PCL utilizando únicamente Visual Studio 2010. Se considera como punto de partida que tenemos instalado tanto Visual Studio 2010 como el paquete completo de la librería PCL. Respecto al paquete de PCL, hay que tener en cuenta que durante la instalación se ha seleccionado que se incluyan automáticamente en la variable PATH del sistema las rutas hacia los elementos de PCL.

En primer lugar, crearemos un proyecto vacío en Visual Studio 2010 dentro del grupo Visual C++. Debemos darle un nombre al proyecto y otro a la solución. En la siguiente figura se puede observar la selección.

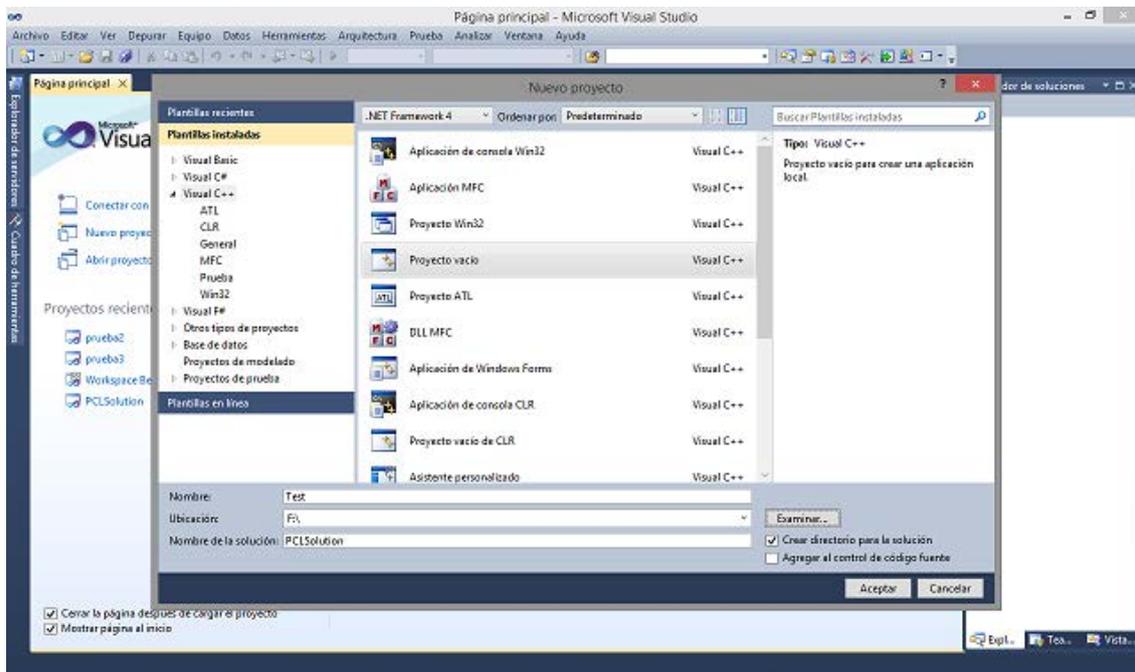


Ilustración 28. Crear un proyecto de PCL con Visual Studio 2010.

El siguiente paso es crear un fichero de código en el grupo “Archivos de código fuente” en el explorador de soluciones. Este fichero debería contener la función main de forma que sirva de punto de entrada a nuestra aplicación. Para este ejemplo, se ha diseñado un programa simple que genera una nube de puntos aleatoria y la visualiza, todo ello utilizando las capacidades de PCL. El código del programa se encuentra en el Anexo IV.

A continuación se debe configurar el entorno de compilación. Esto significa que debemos configurar las dependencias de nuestro software. Para ello, hacemos clic derecho sobre el proyecto en el explorador de soluciones y seleccionamos la opción “Propiedades”. Lo primero que debemos hacer en la ventana que nos aparece es seleccionar en el campo situado arriba a

la izquierda llamado Configuración, el valor “Todas las config.”. Esto se hace para aplicar los cambios que hagamos a todas las configuraciones de compilación que tengamos, que por defecto son Debug y Release. Si bien es cierto que esto es más una decisión propia de cada proyecto y cuya explicación del uso que se le pueda dar no tiene cabida en este documento, se considera que dada la necesidad de incluir todas las dependencias para poder compilar el proyecto, lo mejor es aplicarlo en todas las configuraciones.

Hecho lo anterior, el primer paso consiste en determinar las carpetas en las que se encuentran los ficheros cabecera que vamos a utilizar de librerías externas. En el ejemplo solo se usan los de PCL, incluyéndose todos ellos, pero pueden incluirse más librerías si se hace uso de ellas. Para realizar esta configuración, editamos el siguiente apartado:

Propiedades de configuración > C/C++ > General > Directorios de inclusión adicionales

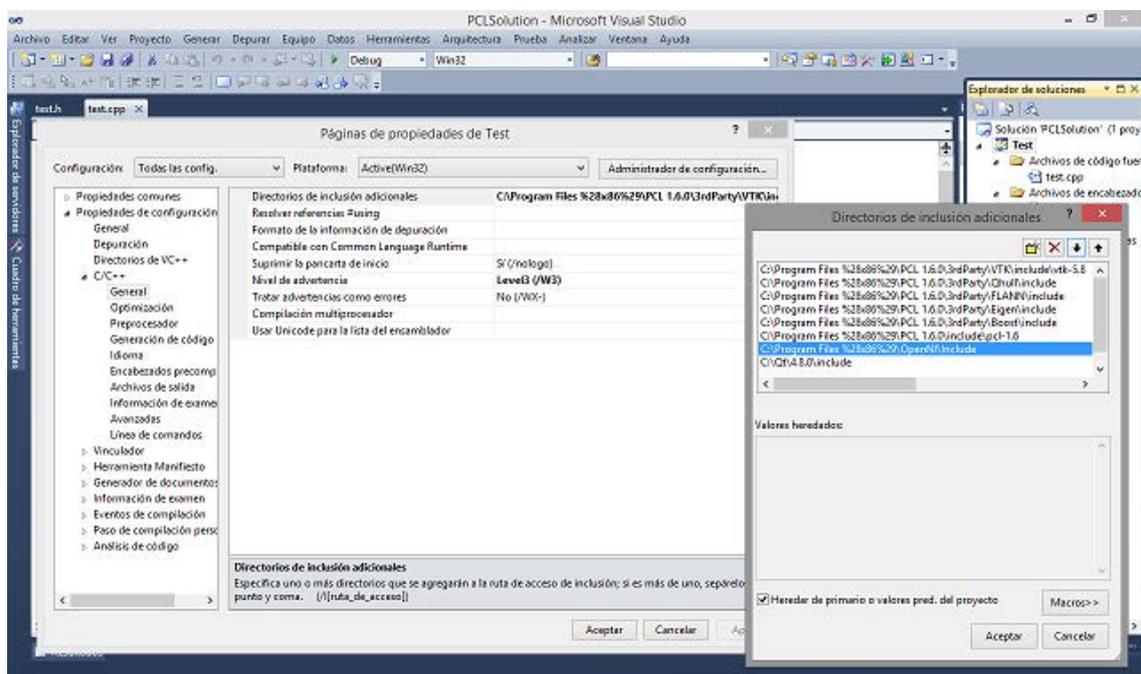


Ilustración 29. Ruta hacia las carpetas include.

En la siguiente tabla se muestran las rutas hacia los ficheros de cabecera necesarios para utilizar PCL en nuestros proyectos. Son las que se incluyen en la ilustración anterior pero de forma más cómoda para leer.

ORIGEN\PCL 1.6.0\include\pcl-1.6
ORIGEN\PCL 1.6.0\3rdParty\VTK\include\vtk-5.8
ORIGEN\PCL 1.6.0\3rdParty\Qhull\include
ORIGEN\PCL 1.6.0\3rdParty\FLANN\include
ORIGEN\PCL 1.6.0\3rdParty\Eigen\include
ORIGEN\PCL 1.6.0\3rdParty\Boost\include
ORIGEN\OpenNI\include
ORIGEN\Qt\4.8.0\include

Tabla 6. Rutas a incluir en PCL.

Hay que considerar que ORIGEN debe sustituirse con la carpeta raíz en la que hayamos instalado las librerías, cosa que queda a elección de cada uno.

El siguiente paso es configurar las carpetas en las que se encuentran los ficheros de librerías utilizables. Igual que en el caso anterior, incluimos todas las rutas de PCL. Esto se puede realizar editando el siguiente campo:

Propiedades de configuración > Vinculador > General > Directorios de bibliotecas adicionales

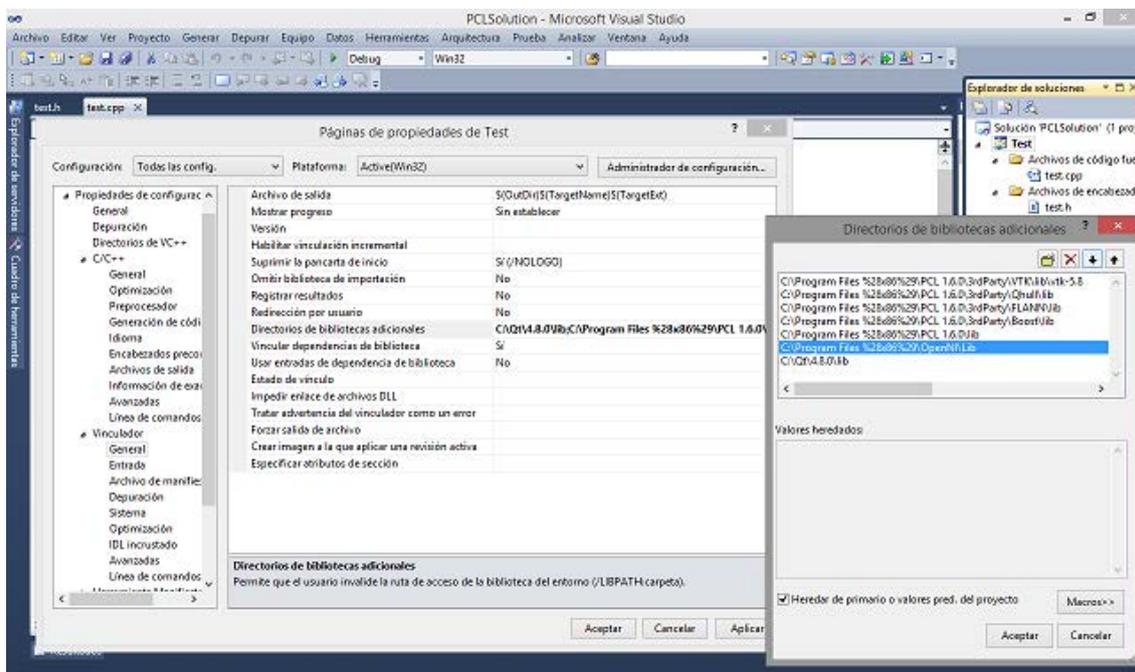


Ilustración 30. Incluir librerías en PCL.

En la siguiente tabla se muestran las rutas hacia las librerías necesarias para utilizar PCL en nuestros proyectos. Son las que se incluyen en la ilustración anterior pero de forma más cómoda para leer.

ORIGEN\PCL 1.6.0\3rdParty\VTK\lib\vtk-5.8
ORIGEN\PCL 1.6.0\3rdParty\Qhull\lib
ORIGEN\PCL 1.6.0\3rdParty\FLANN\lib
ORIGEN\PCL 1.6.0\3rdParty\Boost\lib
ORIGEN\PCL 1.6.0\lib
ORIGEN\OpenNI\Lib
ORIGEN\Qt\4.8.0\lib

Tabla 7. Rutas a las librerías de PCL.

Al igual que en el caso anterior, ORIGEN debe sustituirse con la carpeta raíz en la que hayamos instalado las librerías, cosa que queda a elección de cada uno.

Por último, nos falta elegir que librerías serán utilizadas para compilar nuestro programa. En este caso se ha utilizado solo las necesarias pero en cualquier otro caso deben añadirse todas y cada una de las librerías que serán utilizadas. Esta labor la realizaremos editando el campo:

Propiedades de configuración > Vinculador > Entrada > Dependencias adicionales

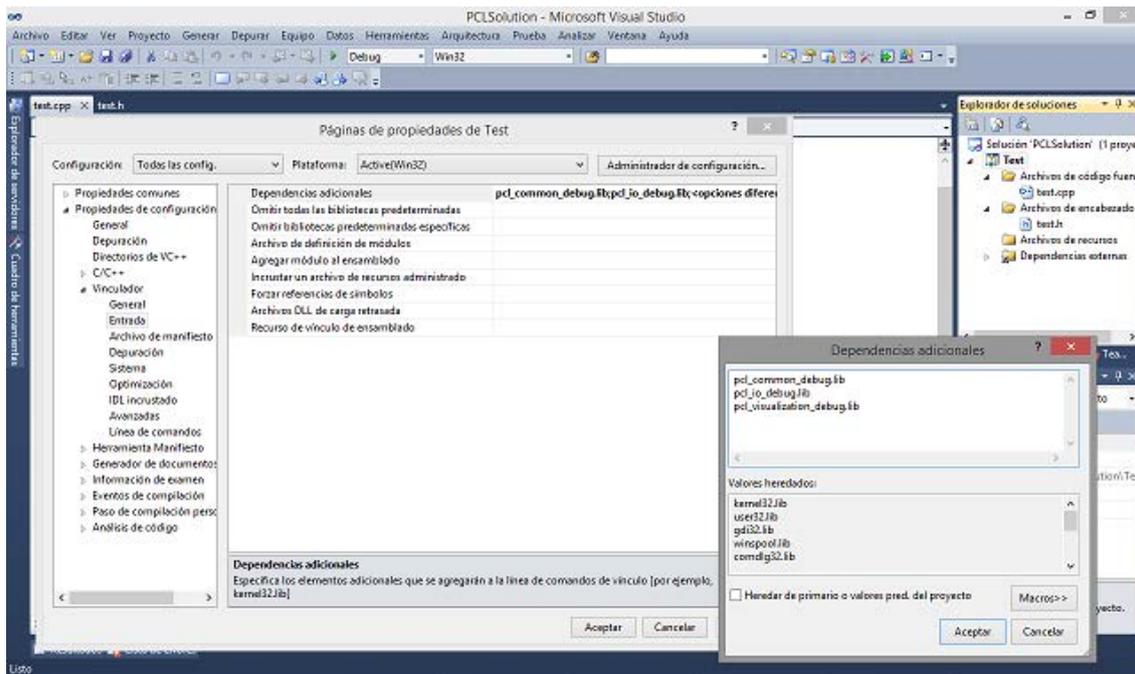


Ilustración 31. Incluir librerías exactas PCL.

En la siguiente tabla se muestran las librerías necesarias para utilizar PCL en este proyecto de ejemplo. Son las que se incluyen en la ilustración anterior pero de forma más cómoda para leer.

pcl_common_debug.lib
pcl_io_debug.lib
pcl_visualization_debug.lib

Tabla 8. Librerías incluidas para el ejemplo PCL.

Estas librerías se corresponden con los módulos “common”, “io” y “visualization”. Si fuera necesario utilizar más funcionalidad, basta con ir a la carpeta lib dentro la instalación de PCL, localizar la librería que necesitamos e incluirla en este apartado de configuración.

Hecho esto hemos terminado de configurar el entorno por lo que si ya tenemos el código programado, podemos ejecutarlo como cualquier otra aplicación desarrollada en Visual Studio. Como se comentó al principio, este ejemplo genera una nube aleatoria de 200 puntos, lo escribe en un fichero y lo visualiza utilizando las capacidades de PCL. El resultado se puede observar en la siguiente ilustración en la que aparece la consola con los puntos generados y la ventana del visualizador en la que se puede apreciar una nube de puntos aleatoria. En dicho visualizador podemos realizar algunas operaciones como mover la cámara o hacer zoom sobre la nube de puntos.

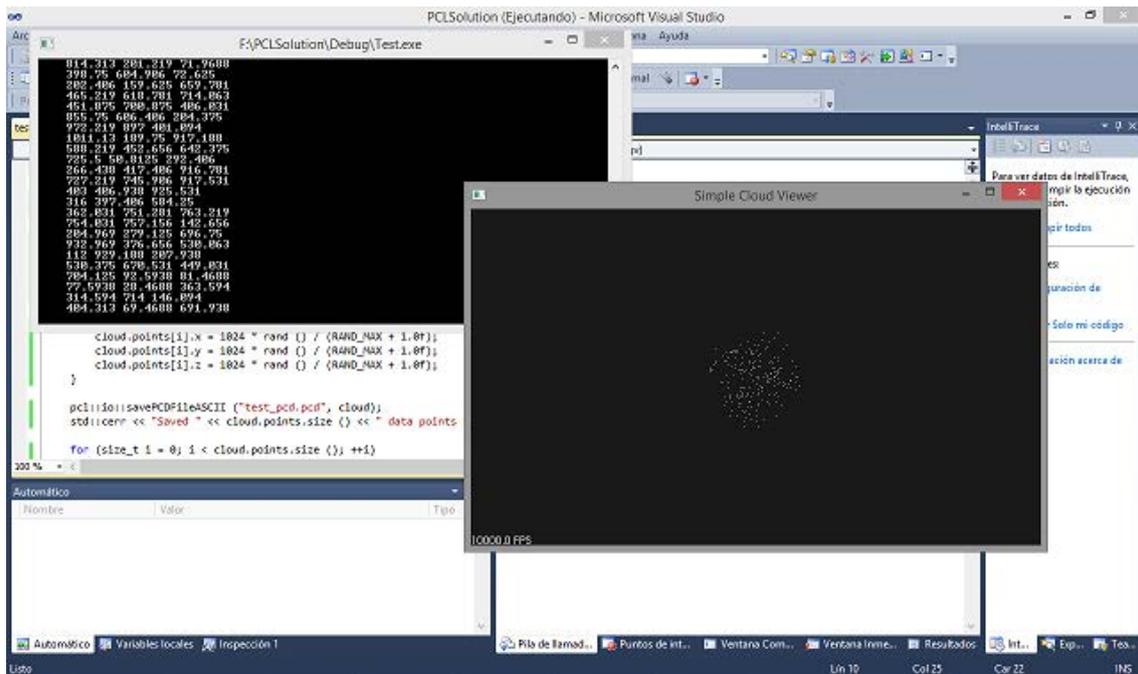


Ilustración 32. Salida del programa de ejemplo de PCL.

A partir de aquí ya se podría programar cualquier aplicación con PCL. Mencionar que este método descrito no abarca todas las posibilidades de la librería por lo que pueden ser requeridas acciones adicionales si utilizamos más funcionalidad, pero dada la magnitud de la librería resultaría demasiado extenso y complejo examinar todas las opciones.

CAPÍTULO 5. PRUEBAS DE CONCEPTO

En este capítulo se describen las diferentes pruebas que se han realizado para valorar la funcionalidad de los elementos analizados así como las desarrolladas específicamente en colaboración con José Manuel Maestre.

CAPÍTULO	5.1 CAPTURAS REALIZADAS
5	
	5.2 PROCESOS CON PCL
	5.3 PROCESOS CON MESHLAB
	5.4 OTROS PROCESOS DE INTERÉS

5.1 Capturas realizadas

En esta sección se describen las diferentes capturas realizadas. En primer lugar, se describe una prueba inicial y posteriormente las pruebas que se realizaron en colaboración con Manuel Sillero, profesor titular en la Facultad de Ciencias de la Actividad Física y del Deporte (INEF-Madrid) y con José Manuel Maestre Rodríguez. Se realizaron un total de tres pruebas, dos de ellas utilizando un maniquí y una última utilizando al propio Manuel Sillero como modelo.

5.1.1 Desarrollo de las capturas

De forma general, la metodología de captura de las diferentes tomas para las pruebas de concepto es la siguiente:

- Se realizó una grabación con la cámara SR4000 de un rostro.
- Se utilizó el software desarrollado para convertir la captura al formato soportado por MeshLab (ASC) para hacer una vista previa del resultado.
- Se utilizó el software desarrollado para filtrar y segmentar la imagen tomada de forma que permaneciese únicamente el rostro y se eliminasen todos aquellos puntos que, debido a su baja confiabilidad u otros motivos, no nos fueran de utilidad para el estudio.
- Se utilizó el software desarrollado para convertir la captura filtrada a los formatos soportados por MeshLab y PCL (ASC y PCD respectivamente) para comprobar de forma visual el resultado del filtrado así como la correcta función de los conversores de formato.

Notar que la primera prueba se describe de forma más exhaustiva mientras que en el resto se omiten ciertos detalles que resultan repetitivos puesto que siguen el mismo proceso que en el primer caso.

5.1.2 Prueba inicial

El objetivo de esta prueba era comprobar que tanto el software desarrollado como el analizado funcionaban correctamente de forma que fuera posible capturar una secuencia, realizar un primer filtrado y convertirla al formato soportado por MeshLab y PCL para su procesamiento posterior. Sirvió a su vez como toma de contacto con las características de la cámara para establecer un marco inicial respecto al procesamiento de los datos obtenidos con la misma.

En primer lugar, se realizó la grabación usando el software proporcionado por la cámara. Una vez realizada la grabación, se procedió a convertir al formato TSF anteriormente mencionado de forma que se pudiera trabajar con él de forma más cómoda utilizando el software de filtrado desarrollado. Se convirtieron 30 frames de la secuencia y tras analizar el conjunto, se tomó



Ilustración 33. Captura de la prueba inicial.

como referencia el frame número 25 dado que se consideró que la imagen capturada y los datos de confiabilidad del mismo eran los más aptos para realizar la prueba. En la ilustración adjunta se observa la imagen correspondiente a los datos de amplitud capturados por la cámara SR4000 en la que se puede ver el rostro que queremos segmentar en primer plano y al fondo el laboratorio de GIAA en el campus de Colmenarejo. El cambio de color de la escena se deba a la distancia del fondo que supera el límite establecido por las características de la cámara para realizar una captura correcta. La siguiente ilustración muestra la nube de puntos en bruto obtenida de la captura.

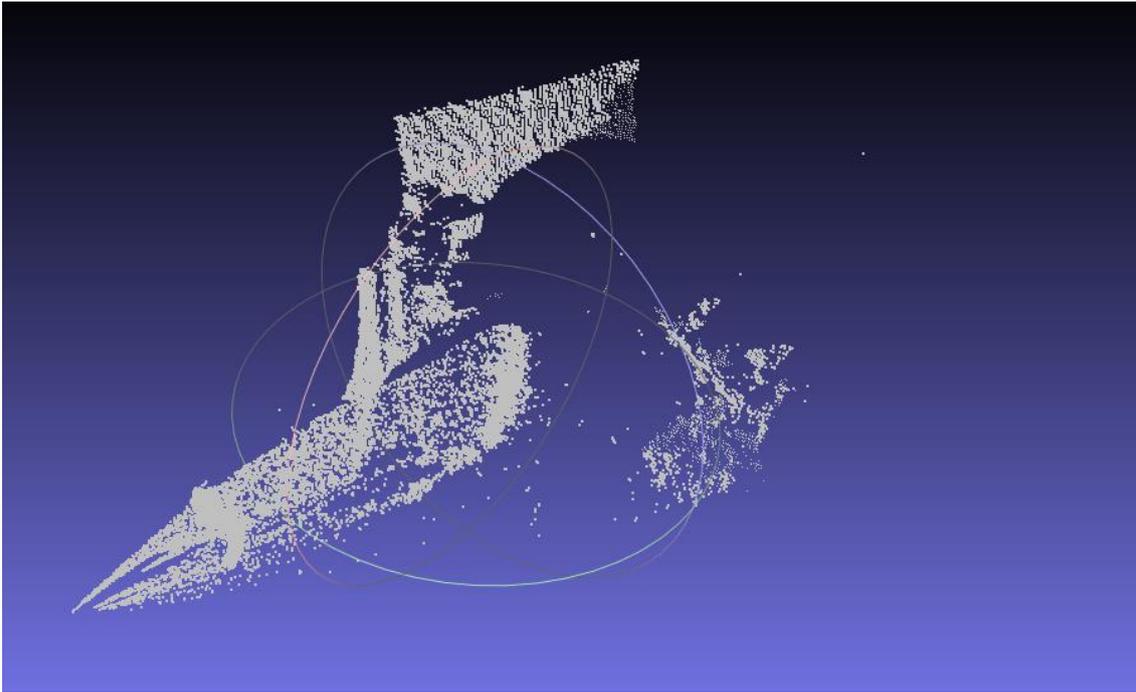


Ilustración 34. Visualización de la prueba inicial sin filtrado.

Se puede observar una especie de cono en la nube de puntos que parte desde el origen hacia un objeto situado en el medio de la escena. Si bien por norma general en las capturas con la cámara SR4000 suele producirse este cono con puntos de baja confiabilidad, en esta captura parece excesivo. Analizando a fondo la captura, se puede observar en la ilustración de la derecha como el problema viene dado por un objeto cuadrado a la derecha del rostro capturado. Dicho objeto se trata de la ventana del fondo cuyo reflejo y excesiva distancia confunde al sensor de la cámara. El hecho de que parezca más cercana de lo que debería es el método utilizado para medir la distancia que utiliza la cámara. Para ello, la cámara analiza el valor de la amplitud de la onda que emite al momento de su recepción.

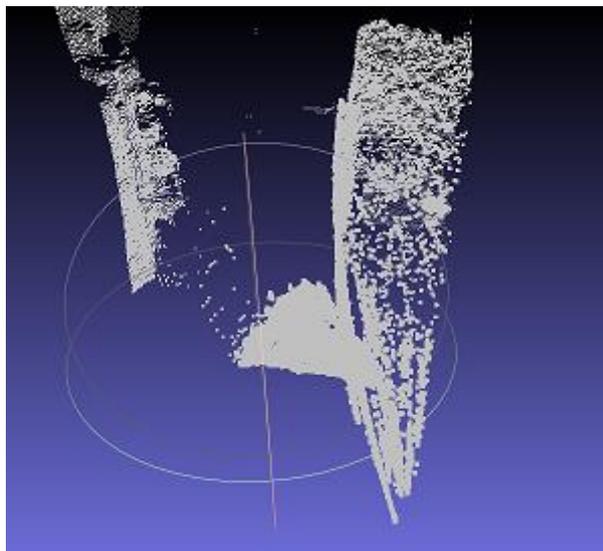


Ilustración 35. Error capturado.

Dado que la onda oscila de forma ininterrumpida, cuando la onda sobrepasa la distancia máxima medible por la cámara, repite sus valores de forma que el sensor, al no saber cuántas veces ha oscilado la onda le asigna el valor de distancia como si solo hubiese oscilado una vez. Sin embargo, los puntos en los que se produce este efecto tienen muy baja confiabilidad por lo que se detectan y eliminan fácilmente.

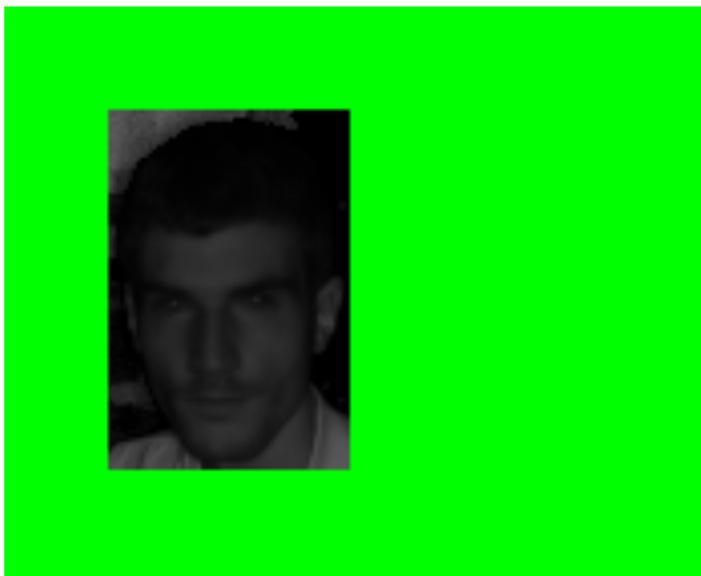


Ilustración 36. Prueba inicial. Aplicación del filtro rectangular.

El siguiente paso realizado fue filtrar la captura para eliminar todos los puntos con baja confiabilidad y segmentar el rostro que es lo que se quería obtener. El proceso de filtrado realizado se detalla a continuación si bien es cierto que cualquier orden en la aplicación de los filtros hubiese supuesto el mismo resultado final.

En primer lugar, se decidió aplicar el filtro rectangular de forma que, con uno de los filtros más simples, se eliminase la mayor parte de los puntos que no se necesitaban para el objetivo de esta prueba. El resultado de aplicar dicho filtro se puede observar en la ilustración de la derecha. Con este filtro se consiguió segmentar la zona del rostro eliminando el resto de la escena capturada, que queda representada con el color verde. Sin embargo, el rostro no está totalmente segmentado ni se han retirado los puntos con baja confiabilidad por lo que se hace necesario seguir aplicando filtros.

El siguiente filtro aplicado es el de confiabilidad. El objetivo de este filtro es eliminar todos aquellos puntos cuyo valor de confiabilidad proporcionado por la cámara es bajo. En este caso



Ilustración 37. Prueba inicial. Aplicación del filtro de confianza.

se decide aplicar un filtro de aproximadamente el 90% de confiabilidad. Dado que, para cada píxel, la cámara establece un valor entre 0 y 65535 siendo a menor valor menor confiabilidad, en este caso se eliminan todos los píxeles cuyo valor sea menor que 58982. El resultado de aplicar este filtro se puede observar en la ilustración de la izquierda. Con este filtro podemos dar por completado el proceso de segmentación del rostro. Se puede observar cómo, a pesar de la cercanía, parte del cabello ha

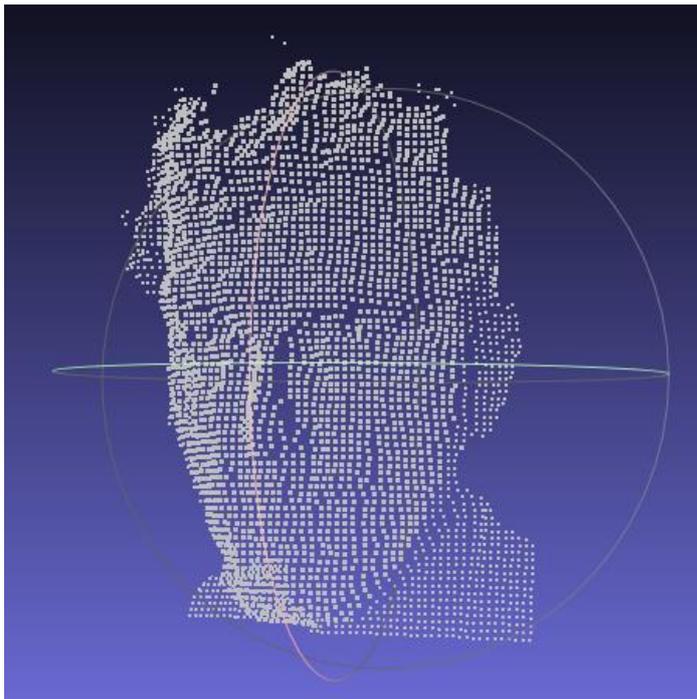


Ilustración 38. Prueba inicial. Visualización final en MeshLab.

sido eliminado. Este problema se ha observado en todas las pruebas realizadas de lo que se concluye que el cabello por sus propiedades produce distorsión en la medida.

Para finalizar esta prueba inicial, se realiza el proceso de generación de ficheros compatibles con MeshLab y PCL. Para ello se utiliza el frame ya filtrado con anterioridad.

En primer lugar, se realiza la conversión al formato ASC compatible con MeshLab. En el segundo proceso de esta prueba ya se realizó dicha conversión para obtener una visualización previa de la captura obtenida por lo que se pudo comprobar que el

conversor funciona adecuadamente. En este caso, se realiza la conversión de la captura ya filtrada y se puede observar el resultado en la ilustración de la izquierda. Como podemos ver, el conversor ha seleccionado correctamente los datos filtrados en el proceso anterior dejando únicamente los puntos pertenecientes al rostro que era el objetivo de esta prueba.

Por otro lado, se realiza también la conversión al formato PCD perteneciente a la librería PCL para comprobar su correcto funcionamiento. El resultado se puede observar en la ilustración de la derecha, conseguida mediante el módulo de visualización incluido en la propia librería PCL y programando una pequeña prueba. Se puede comprobar como el resultado es idéntico para ambas herramientas por lo que se concluye que el conversor al formato PCD también funciona correctamente.

5.1.3 Captura del maniquí completo

Para esta prueba se utilizó un maniquí, capturado a cuerpo completo. Se realizaron un total de ocho capturas sobre el maniquí de forma que se lograra capturar toda



Ilustración 39. Prueba inicial. Visualización final en PCL.

la superficie del mismo. En la siguiente ilustración se puede observar el maniquí objeto de la captura.



Ilustración 40. Segunda prueba de concepto. Primer caso.

Una vez realizadas todas las capturas, el siguiente paso es el filtrado de cada una de ellas para quedarnos únicamente con los datos de valor. Para ello, se elimina el fondo y todos aquellos puntos que tienen baja confiabilidad. En la siguiente secuencia de imágenes se puede observar el filtrado de todas las capturas realizadas. Dado que el proceso de filtrado ya lo hemos visto en la primera prueba de concepto, vamos a obviar el desarrollo del mismo ya que se utiliza la misma herramienta y de la misma forma. Dicho esto, en la siguiente página se pueden observar los resultados finales de filtrado de cada toma.

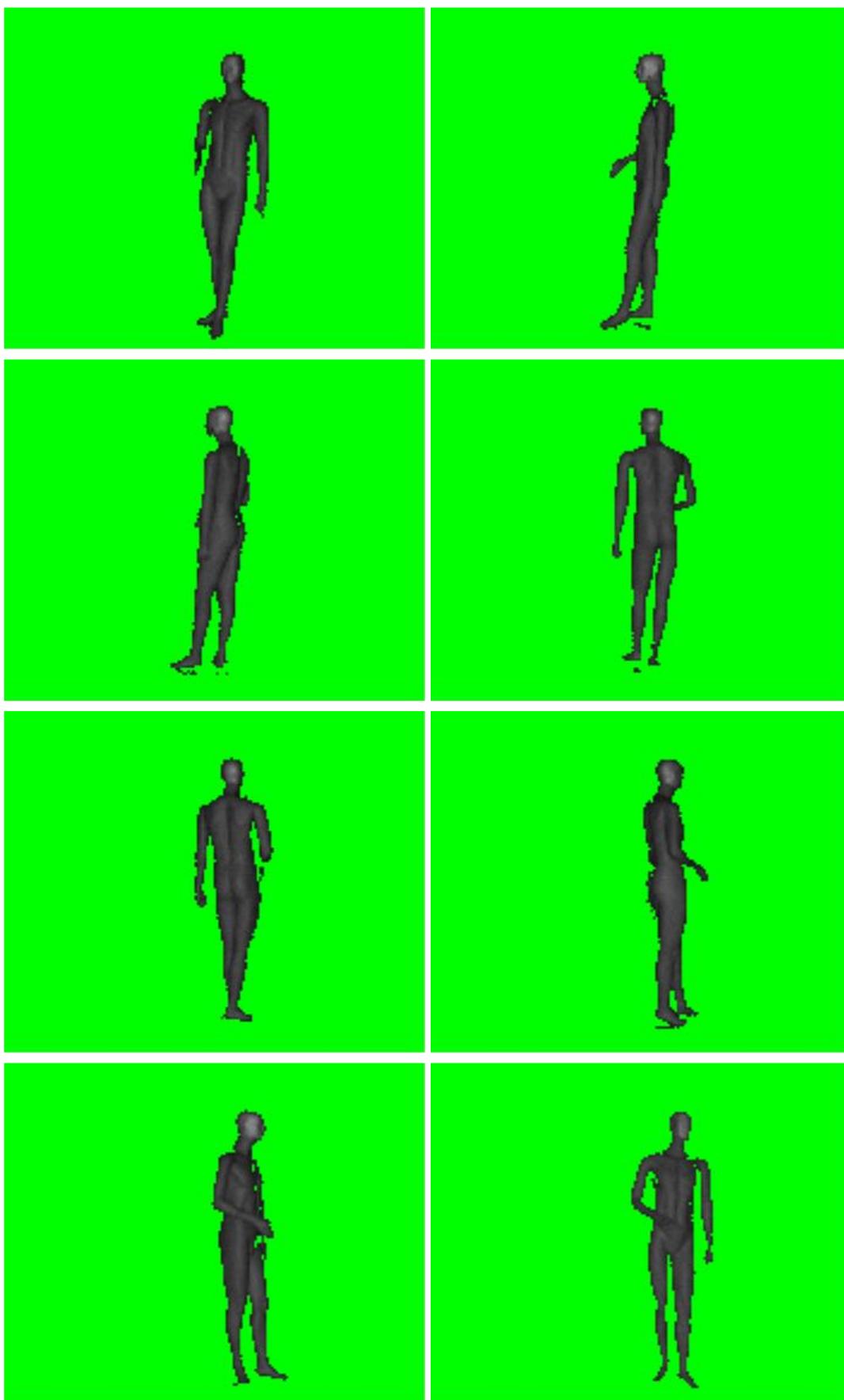


Ilustración 41. Secuencia A filtrada.

Una vez que tenemos las diferentes capturas filtradas, se exportan a nubes de puntos. En este caso, para visualizar el resultado utilizamos el programa MeshLab y tomamos una imagen de la primera captura que podemos ver en la siguiente ilustración. A priori parece ser correcta pero manipulando el punto de vista se pueden observar diversas anomalías sobre la superficie.

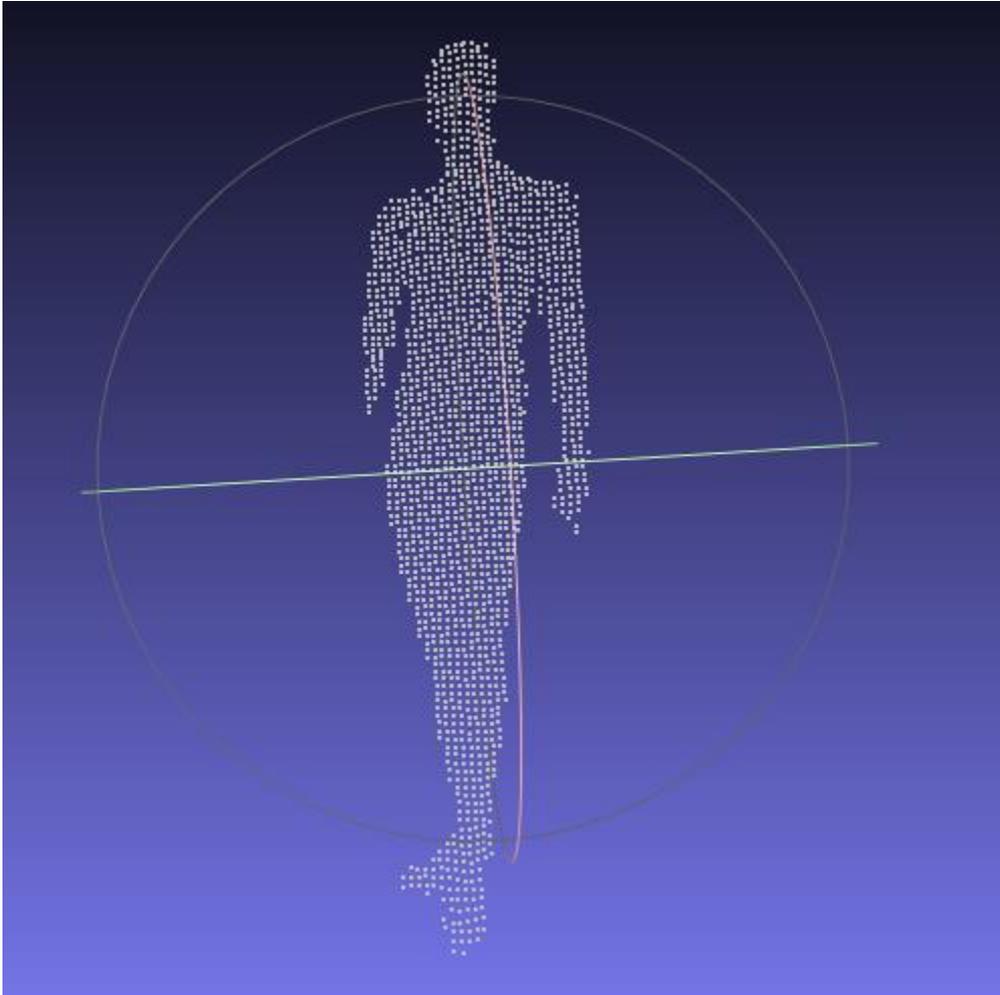


Ilustración 42. Nube de puntos en el caso A.

En cualquier caso, a pesar de que la superficie del objeto no parezca la correcta, el perfil del mismo parece más o menos acertado por lo que se pueden realizar unas primeras pruebas de fiabilidad en la precisión de la cámara. Utilizando las herramientas de MeshLab realizamos la medición de la altura del objeto dando como resultado aproximadamente 170 centímetros, lo cual concuerda con el objeto real. De este resultado podemos deducir que al menos la medición que realiza la cámara es correcta.

5.1.4 Captura del torso del maniquí

Para esta prueba se utilizó un maniquí, capturando solamente el torso del mismo y a menor distancia que la prueba anterior. Al igual que en la primera prueba, se realizaron un total de ocho capturas sobre el maniquí de forma que se lograra capturar toda la superficie del mismo. En la siguiente ilustración se puede observar la parte del maniquí objeto de la captura.



Ilustración 43. Torso del maniquí.

Una vez realizadas todas las capturas, el siguiente paso es el filtrado de cada una de ellas para quedarnos únicamente con los datos de valor. Para ello, se elimina el fondo y todos aquellos puntos que tienen baja confiabilidad. En la siguiente secuencia de imágenes se puede observar el filtrado de todas las capturas realizadas. Dado que el proceso de filtrado ya lo hemos visto en la primera prueba de concepto, vamos a obviar el desarrollo del mismo ya que se utiliza la misma herramienta y de la misma forma. Dicho esto, en la siguiente página se pueden observar los resultados finales de filtrado de cada toma.

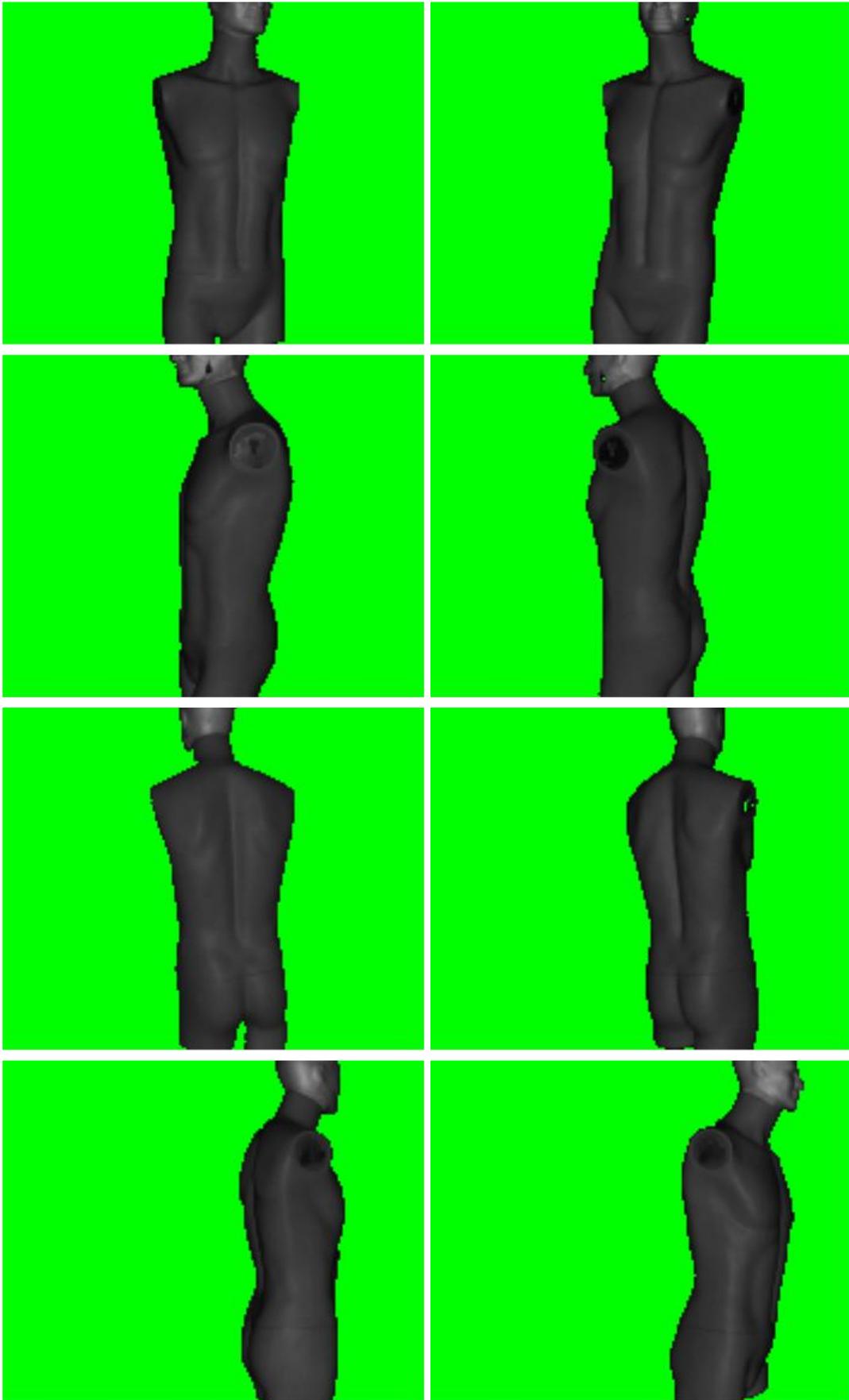


Ilustración 44. Secuencia B filtrada.

Una vez que tenemos las diferentes capturas filtradas, se exportan a nubes de puntos. En este caso, para visualizar el resultado tomamos una imagen de la primera captura que podemos ver en la siguiente ilustración. Sucede lo mismo que en el caso anterior, las proporciones y disposición de los puntos en la nube es correcta pero se siguen observando diversas anomalías en la superficie del objeto.

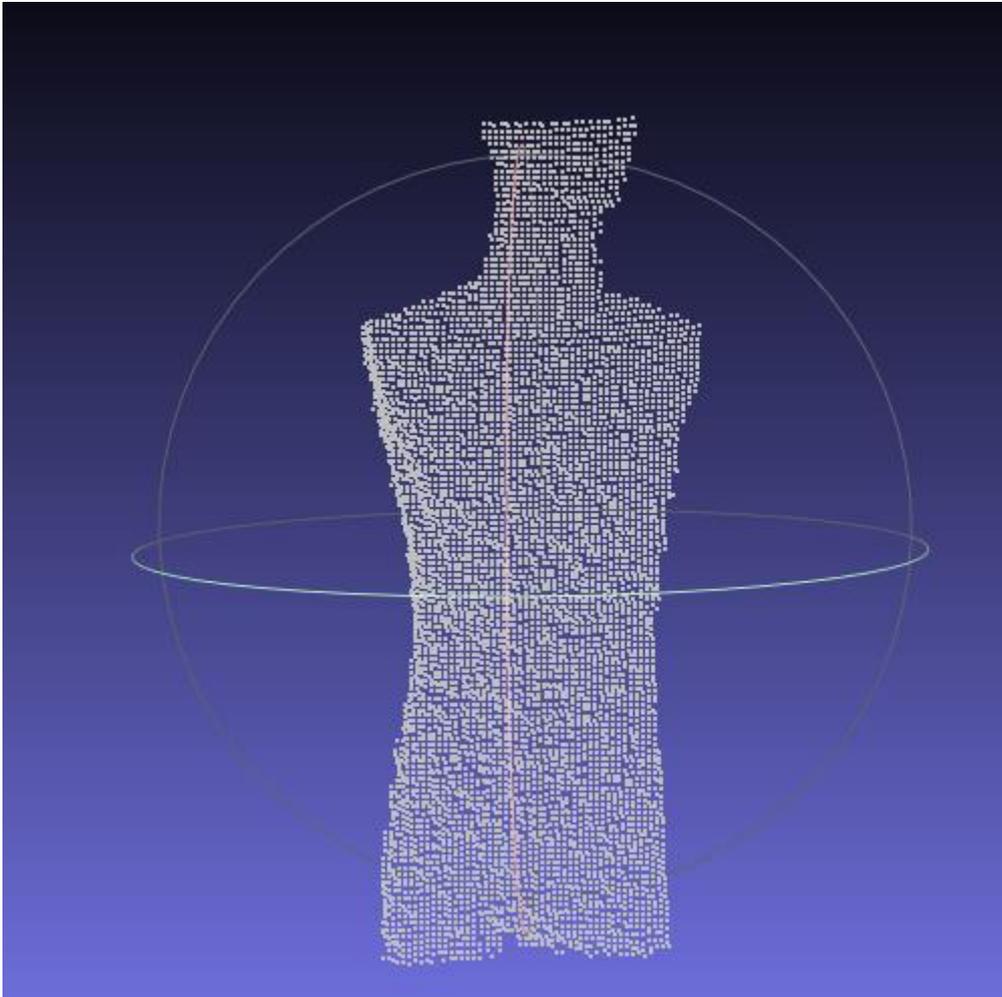


Ilustración 45. Nube de puntos del torso del maniquí.

5.1.5 Captura de Manuel Sillero

Para esta prueba se prestó a hacer de modelo el propio Manuel Sillero. En este caso se realizó una captura a cuerpo completo, muy similar a la primera prueba con el maniquí. Al igual que en dicha primera prueba, se realizaron un total de ocho capturas sobre Manuel de forma que se lograra capturar toda la superficie posible y eliminar los posibles huecos. En la siguiente ilustración se puede observar el modo en que se realizaron las capturas.



Ilustración 46. Manuel Sillero posando para la tercera prueba.

Una vez realizadas todas las capturas, el siguiente paso es el filtrado de cada una de ellas para quedarnos únicamente con los datos de valor. Para ello, se elimina el fondo y todos aquellos puntos que tienen baja confiabilidad. En la siguiente secuencia de imágenes se puede observar el filtrado de todas las capturas realizadas. Todas ellas han sido filtradas utilizando la misma metodología de filtrado. Por un lado se ha aplicado el filtro rectangular para seleccionar el área que contiene el modelo y posteriormente se ha aplicado el filtro de confiabilidad alrededor del 80% y se ha jugado con los filtros de distancia frontal y trasera para quedarnos únicamente con el modelo. Dado que el proceso de filtrado ya lo hemos visto en la primera prueba de concepto, vamos a obviar el desarrollo del mismo ya que se utiliza la misma herramienta y de la misma forma. Dicho esto, en la siguiente página se pueden observar los resultados finales de filtrado de cada toma.

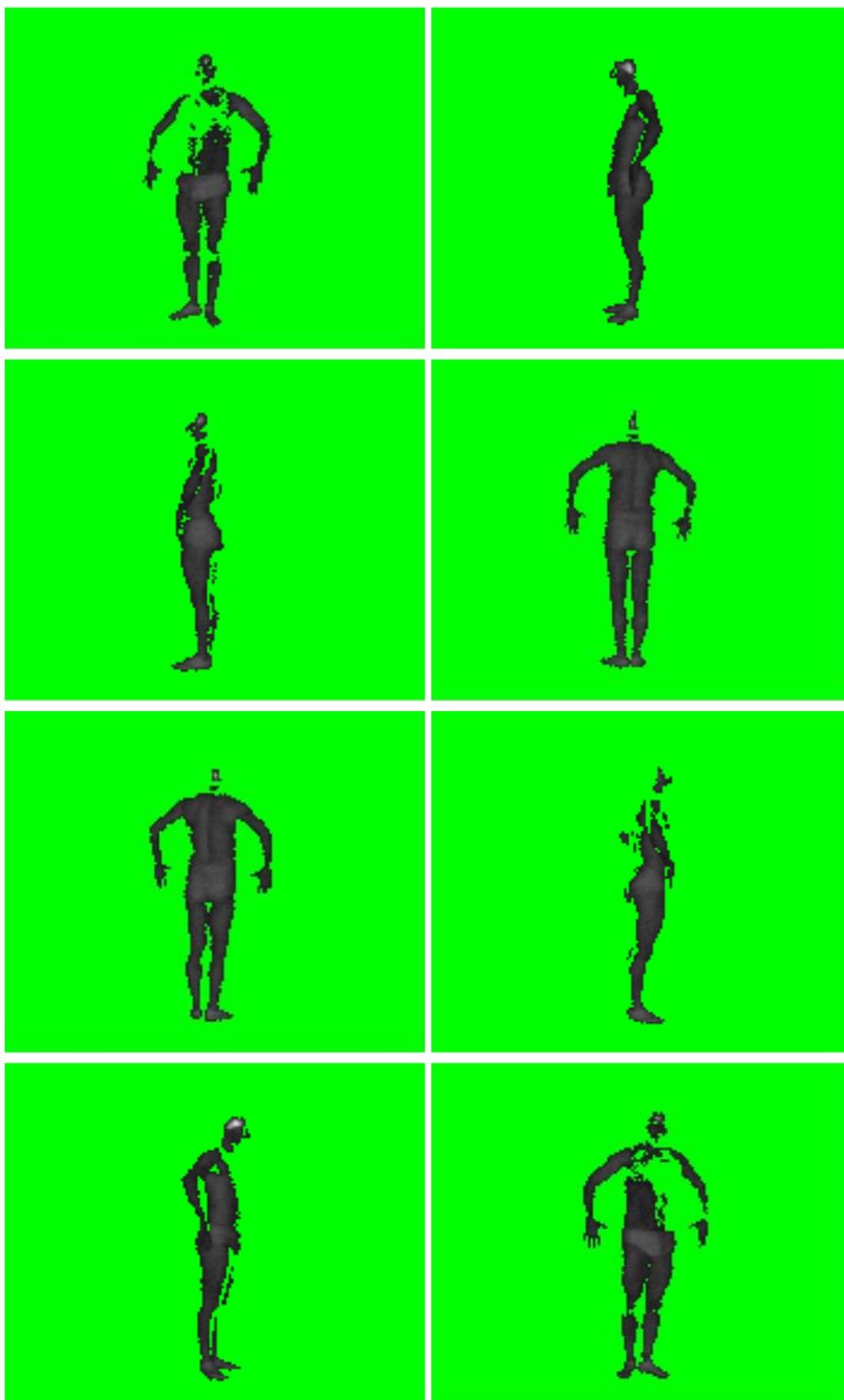


Ilustración 47. Secuencia de Manuel Sillero filtrada.

En este caso, como ya se puede observar en la secuencia filtrada existen muchos vacíos fundamentalmente en capturas frontales sobre la zona del torso. Esto es debido, como ya se ha comentado con anterioridad, a la distorsión que produce el cabello o vello corporal sobre la captura. En la siguiente ilustración se puede observar el modelo sin filtrado de confiabilidad de forma que se puede ver la distorsión producida. Se ha filtrado de forma manual utilizando MeshLab para mantener únicamente el torso, que es donde se concentra principalmente el error de captura.

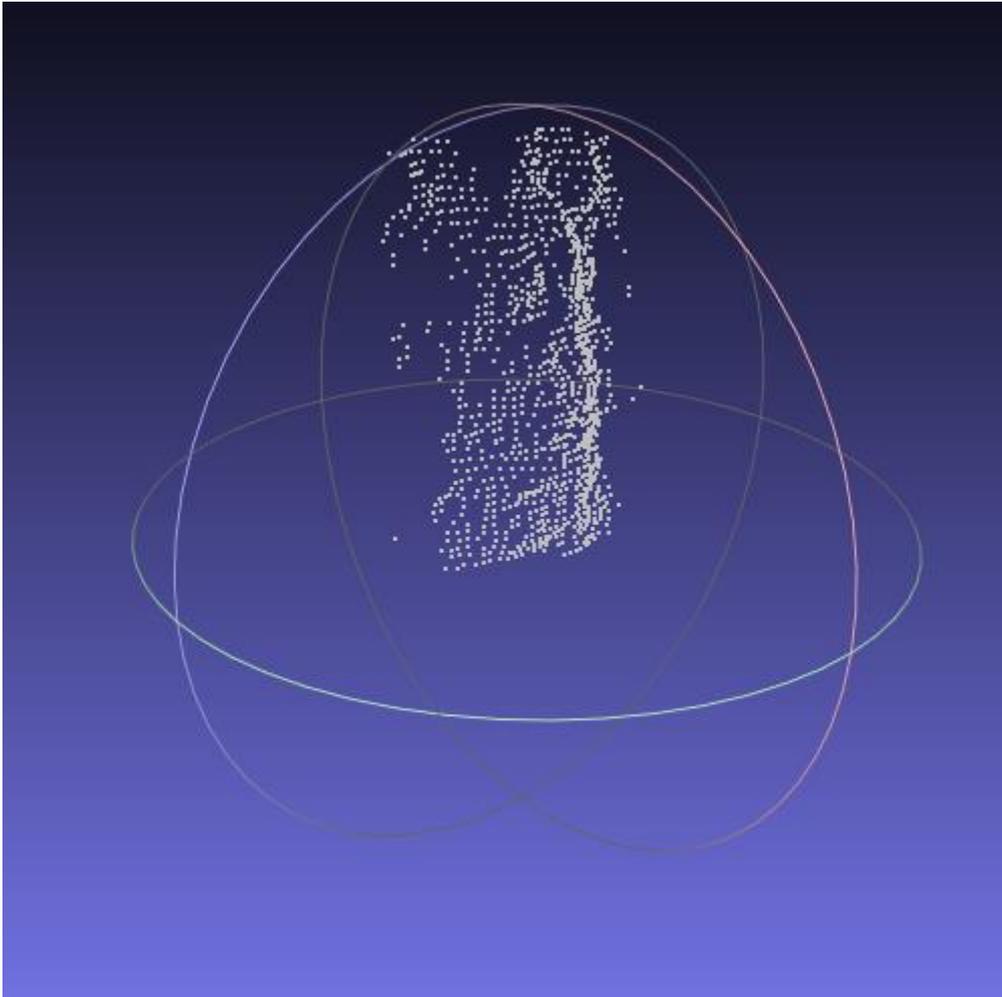


Ilustración 48. Fallo en la captura del torso.

Es complicado observarlo en una imagen estática pero, si bien en todos los casos existe cierto grado de error sobre la captura de la superficie, en este caso su magnitud es mucho más notable.

5.2 Procesos con PCL

Una vez que tenemos realizadas todas las capturas procedemos a ejecutar diferentes procesos sobre los datos utilizando la herramienta PCL. Todos los procesos mostrados a continuación se encuentran disponibles en el fichero de código Test.cpp incluido en el anexo correspondiente.

5.2.1 Estado inicial

Para realizar las pruebas con PCL se toma como punto de partida la siguiente nube de puntos obtenida de la captura del torso del maniquí. Dicha captura ya ha sufrido un filtrado inicial (detallado anteriormente) por lo que el conjunto de puntos inicial es el que se puede observar en la siguiente ilustración.

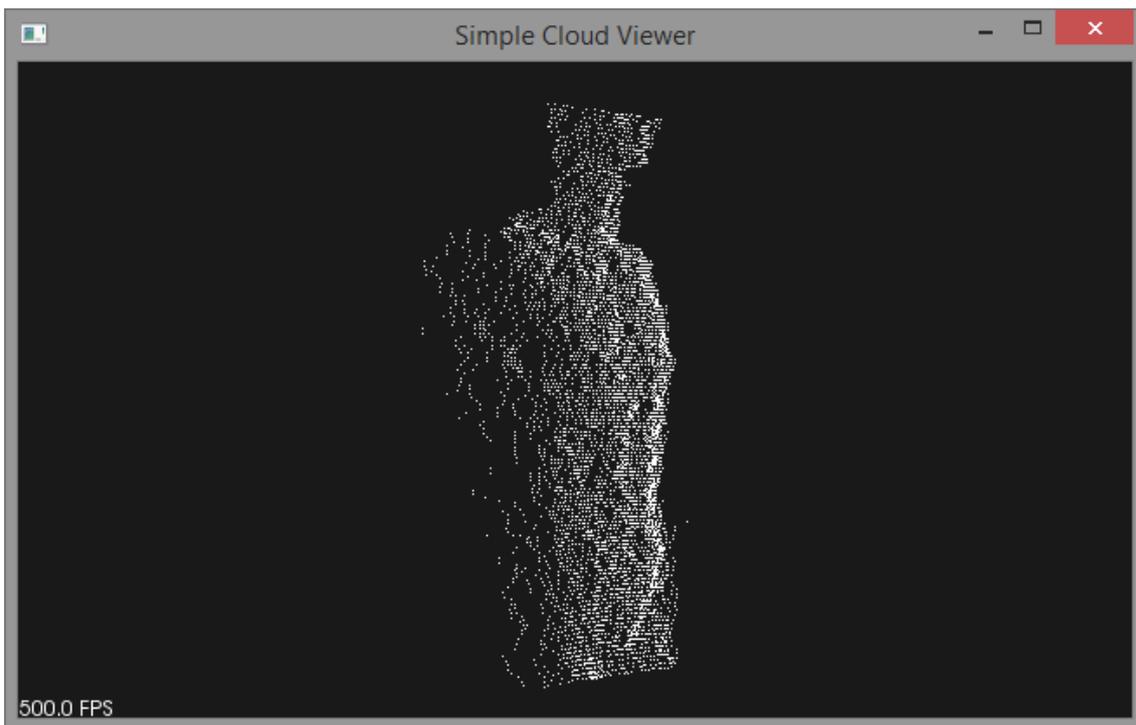


Ilustración 49. Estado inicial en la prueba con PCL.

Como se puede observar, existe cierta irregularidad en la superficie del objeto capturado así como puntos que aportan poca información o incluso errónea que son los situados en el lateral del maniquí.

5.2.2 Filtrado de la nube de puntos

El primer proceso que se aplica es el filtrado de la nube de puntos. El objetivo de este procedimiento es doble. Por un lado, se busca eliminar aquellos puntos que aportan poca información al conjunto, fundamentalmente debido a que se encuentran aislados del resto. Por otra parte, es un método de eliminación de errores de captura pues suele ser habitual que dichos errores generen puntos aislados del resto que pueden ser eliminados mediante este tipo de filtros.

En este caso se utilizará un filtro estadístico para eliminar puntos aislados. La idea de este filtro implementado en PCL es realizar un análisis estadístico de la vecindad de cada punto y eliminar aquellos que no cumplen un determinado criterio. Este caso en particular se basa en calcular la distribución de distancias entre un punto y sus vecinos. Para cada punto se calcula la distancia

media entre este y todos sus vecinos y asumiendo que el resultado es una distribución Gaussiana con una media y una desviación estándar determinadas, todos los puntos que queden fuera del intervalo definido por estos parámetros son eliminados.

El resultado de aplicar este filtro se puede observar en la siguiente ilustración.

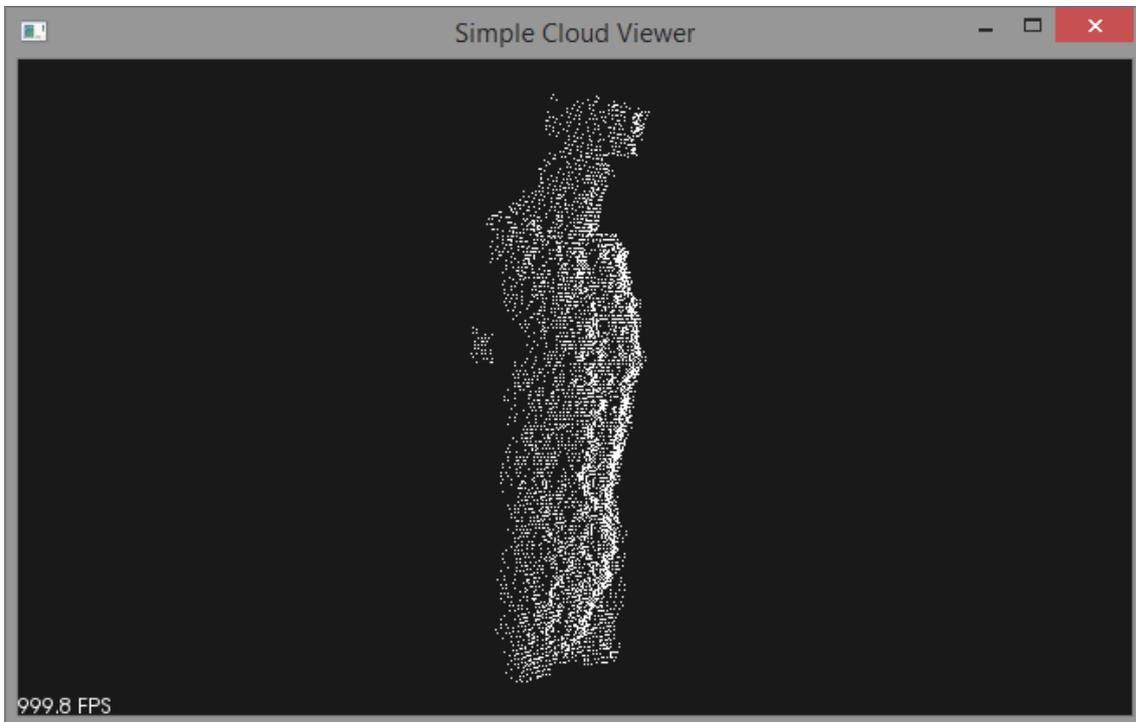


Ilustración 50. Modelo de la prueba con PCL filtrado.

Como se puede observar, han desaparecido gran parte de los puntos del lateral de la captura debido a que la cámara ha sido capaz de captar menos puntos en esta zona y consecuentemente eran puntos más aislados que el resto. También se puede observar como algún punto del frontal que parecía un error de captura ha sido eliminado.

5.2.3 Suavizado de la nube de puntos

El segundo proceso que se aplica es el suavizado de la nube de puntos. Se puede observar en la ilustración anterior como la superficie capturada parece contener mucho ruido y se asemeja poco a la superficie lisa del maniquí. Para corregir este problema se aplica un proceso de suavizado cuya función es reducir en la medida de lo posible la distorsión que se produce durante la captura del modelo.

Muchas de las irregularidades causadas por errores de medición pequeños son muy difíciles de eliminar usando análisis estadístico. Una solución es utilizar algún algoritmo de remuestreo cuya función es recrear partes faltantes de la superficie utilizando interpolaciones polinómicas de alto orden entre los puntos de los alrededores. Está demostrado que con este remuestreo, los pequeños errores superficiales pueden ser corregidos así como otros problemas como el de “dobles paredes” resultante de la alineación de varios escaneos.

En este caso se utilizará el método Robust Moving Least Squares (RMLS) implementado en PCL. Existen muchos métodos que utilizan los principios básicos de MLS como los mencionados en el artículo *A Survey of Methods for Moving Least Squares Surfaces* (16).

El resultado de aplicar este algoritmo se puede observar en la siguiente ilustración.

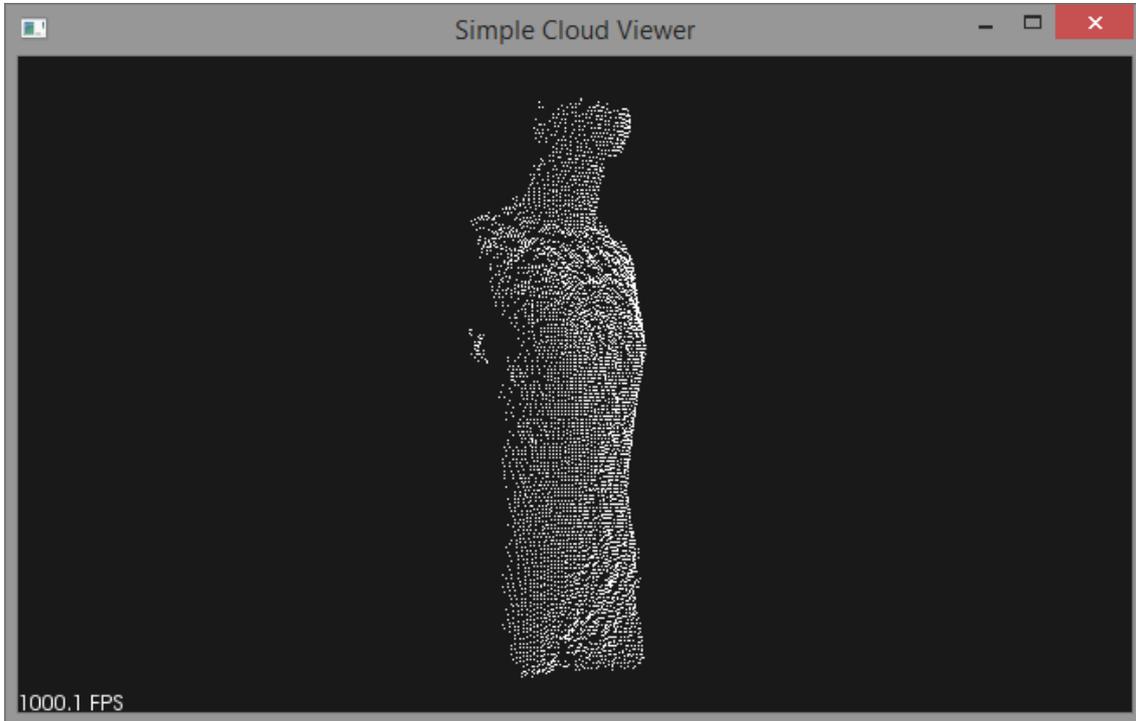


Ilustración 51. Modelo de la prueba con PCL suavizado.

Se puede observar como la superficie del modelo ha quedado mucho más lisa que en el paso anterior y tiende a asemejarse al modelo original. Es cierto que en este caso ha funcionado correctamente el algoritmo pero es necesario determinar en cada caso si es necesario aplicarlo o no. Por ejemplo, si escaneamos una superficie irregular es probable que aplicar este procedimiento sea inútil e incluso contraproducente.

5.2.4 Triangulación de la nube de puntos

El último paso que aplicaremos es la triangulación del modelo cuyo objetivo es recrear la superficie del mismo utilizando triángulos cuyos vértices serán los puntos de la nube.

Existen numerosos algoritmos para realizar la triangulación de una nube de puntos como los derivados de Delaunay o Voronoi. En este caso se ha optado por utilizar un algoritmo de triangulación voraz implementado en PCL. El mismo funciona manteniendo una lista de puntos de los cuales el modelo pueda crecer extendiéndolo hasta que todos los puntos posibles están conectados.

En la siguiente ilustración podemos observar la triangulación del modelo anterior. Para visualizarla se ha utilizado el programa ParaView dado que es capaz de mostrar directamente la información de triangulación arrojada por PCL.

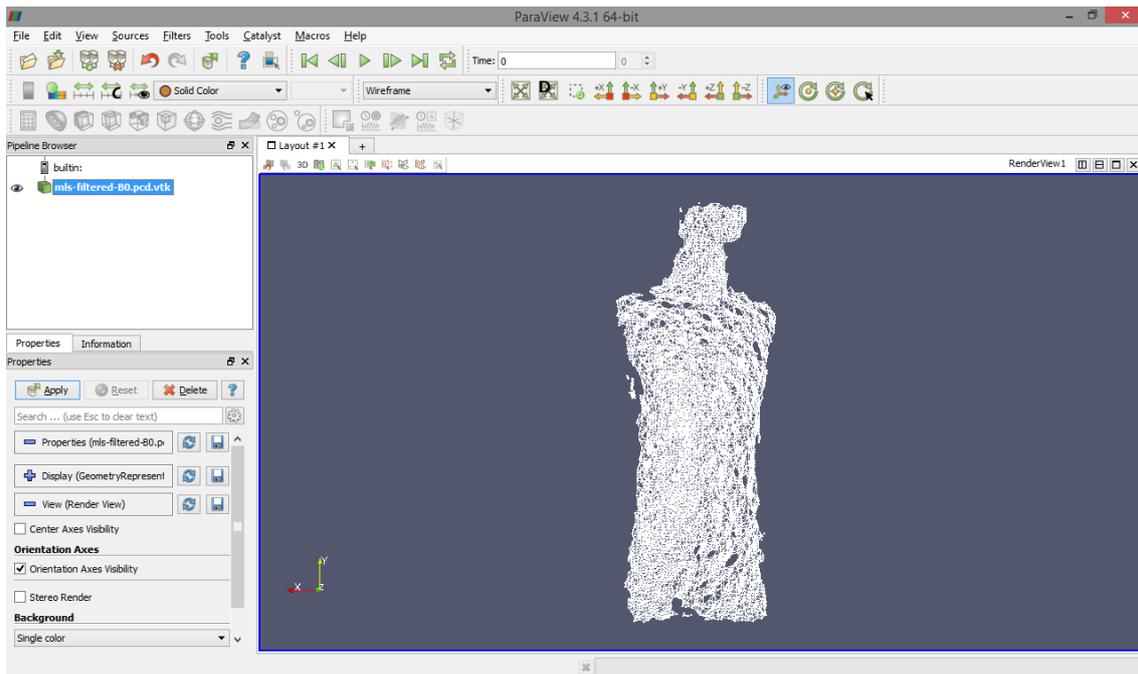


Ilustración 52. Modelo de la prueba con PCL triangulado.

El resultado observable es una malla de triángulos que representa la superficie del modelo escaneado.

5.3 Procesos con MeshLab

Al igual que en el caso anterior, en este apartado se describirán diversos procesos sobre las nubes de puntos pero esta vez utilizando el programa MeshLab.

5.3.1 Alineado manual de nubes de puntos

Para realizar esta prueba con Meshlab se toma como captura de trabajo la realizada sobre el cuerpo completo del maniquí. Dado que el objetivo es alinear varias nubes de puntos, se seleccionaron cuatro de ellas sobre las que se realizó un filtrado inicial. En la siguiente ilustración se muestra una de las nubes de puntos seleccionadas.

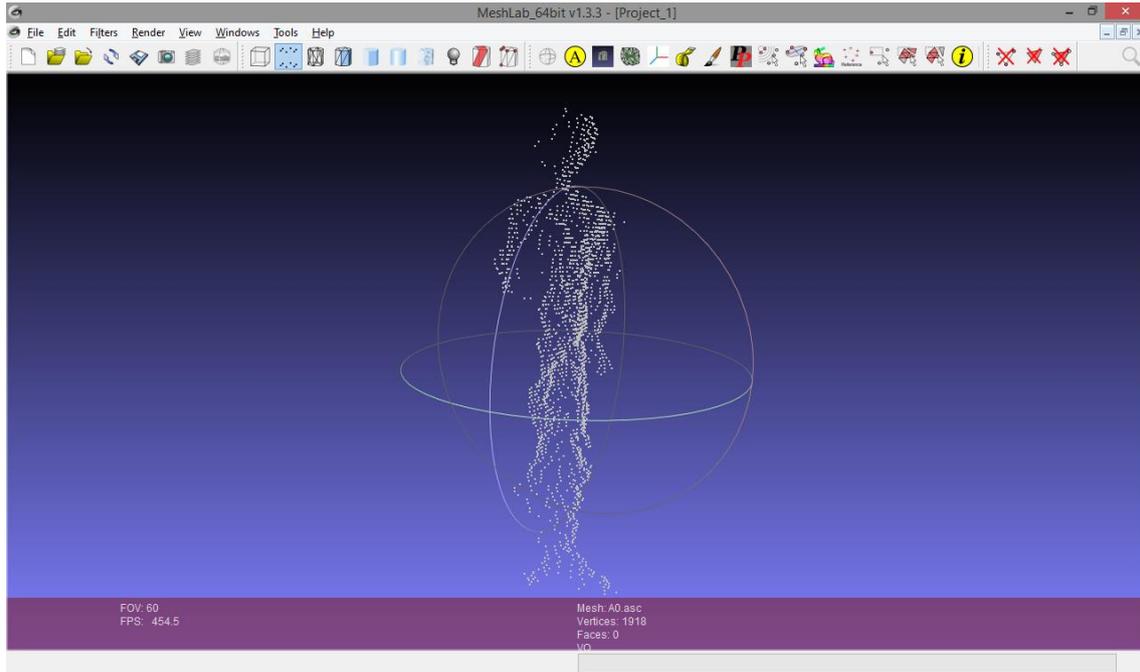


Ilustración 53. Captura inicial para las pruebas con MeshLab.

Para realizar la alineación manual, hay que seleccionar la herramienta Align (icono amarillo con una A) y seleccionar la primera de las nubes de puntos. Hecho esto se pulsará sobre el botón Glue mesh here, estableciendo esta nube de puntos como base.

Posteriormente, seleccionamos la siguiente de las nubes de puntos en la herramienta de alineado y pulsamos sobre Point Based Glueing. Esta herramienta nos permitirá seleccionar un mínimo de 4 puntos concordantes entre ambas nubes con las que se realizarán las transformaciones oportunas para alinear ambas nubes. En la siguiente ilustración se puede observar el proceso.

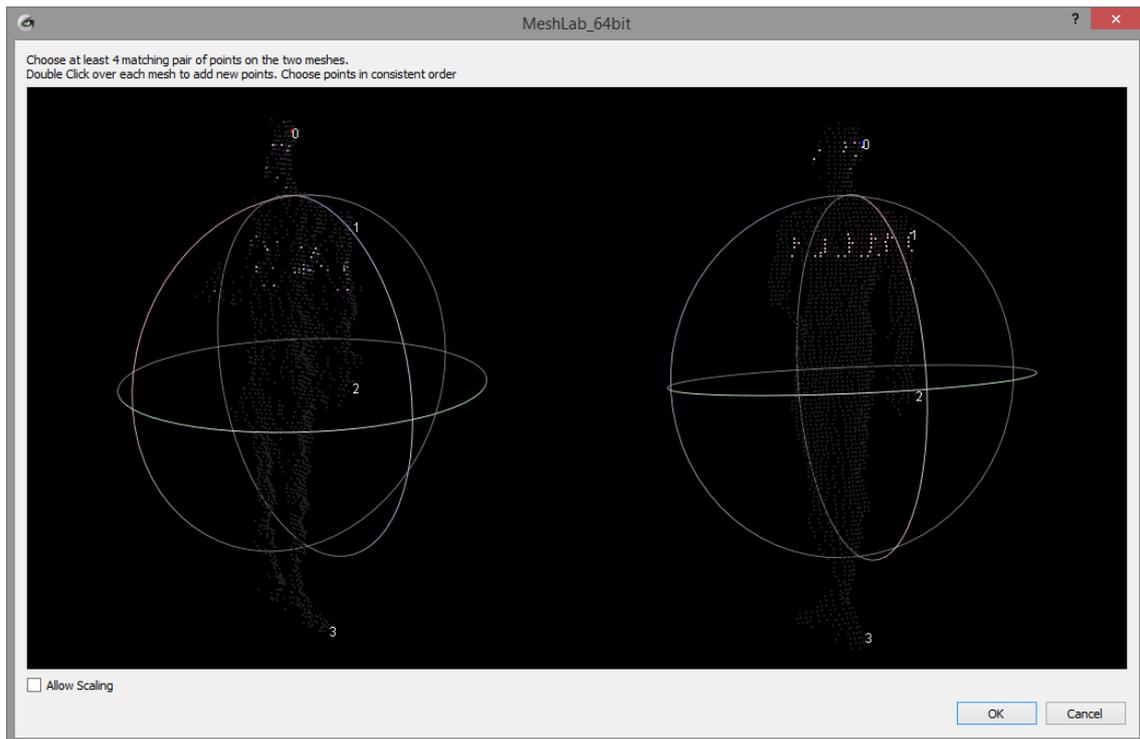


Ilustración 54. Selección de puntos para alinear.

A pesar de no requerir que los puntos seleccionados sean exactamente concordantes, el resultado será mejor cuanto mayor aproximación haya. Por otro lado, la orientación de las nubes en la herramienta se utiliza como estimación inicial por lo que es conveniente orientarlas de forma adecuada. Esta operación hay que repetirla por cada nube que se quiera alinear. En la siguiente serie de ilustraciones podemos observar el estado inicial de las cuatro nubes de puntos y el resultado final desde dos perspectivas.

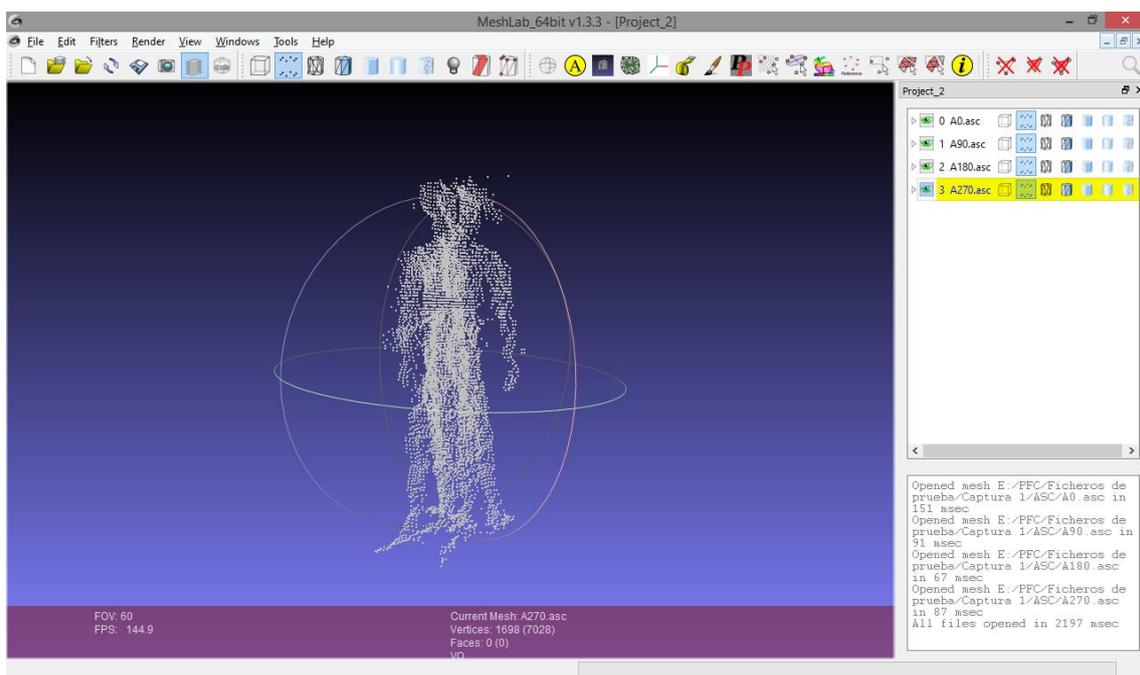


Ilustración 55. Estado inicial de las nubes de puntos.

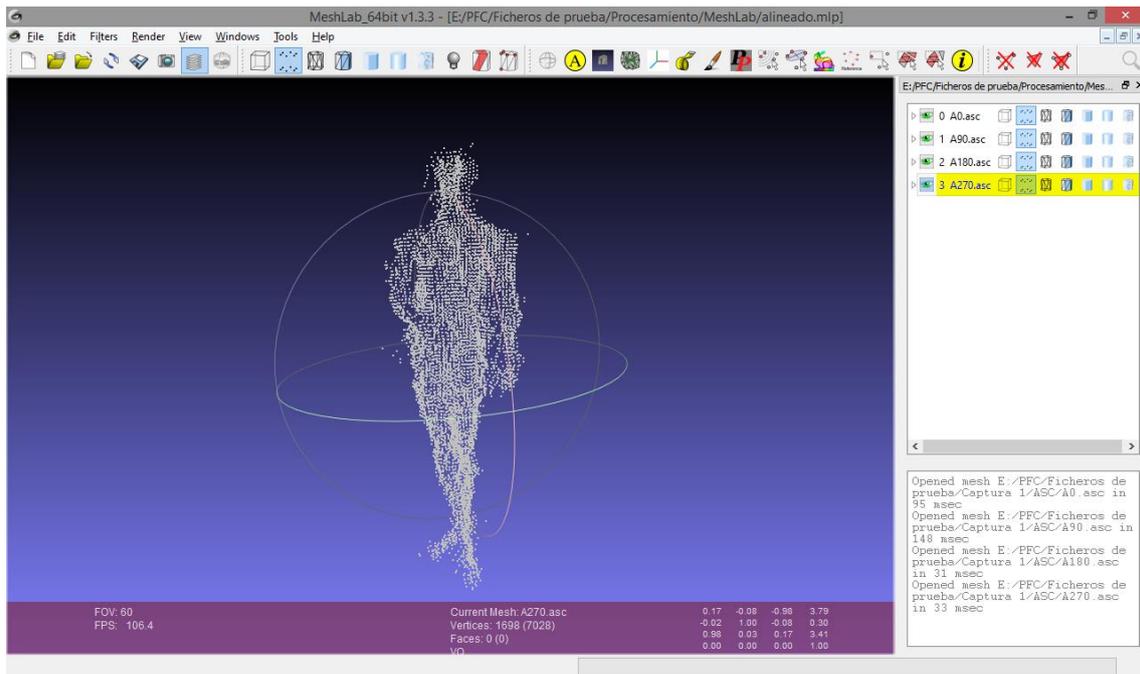


Ilustración 56. Estado final, perspectiva 1.

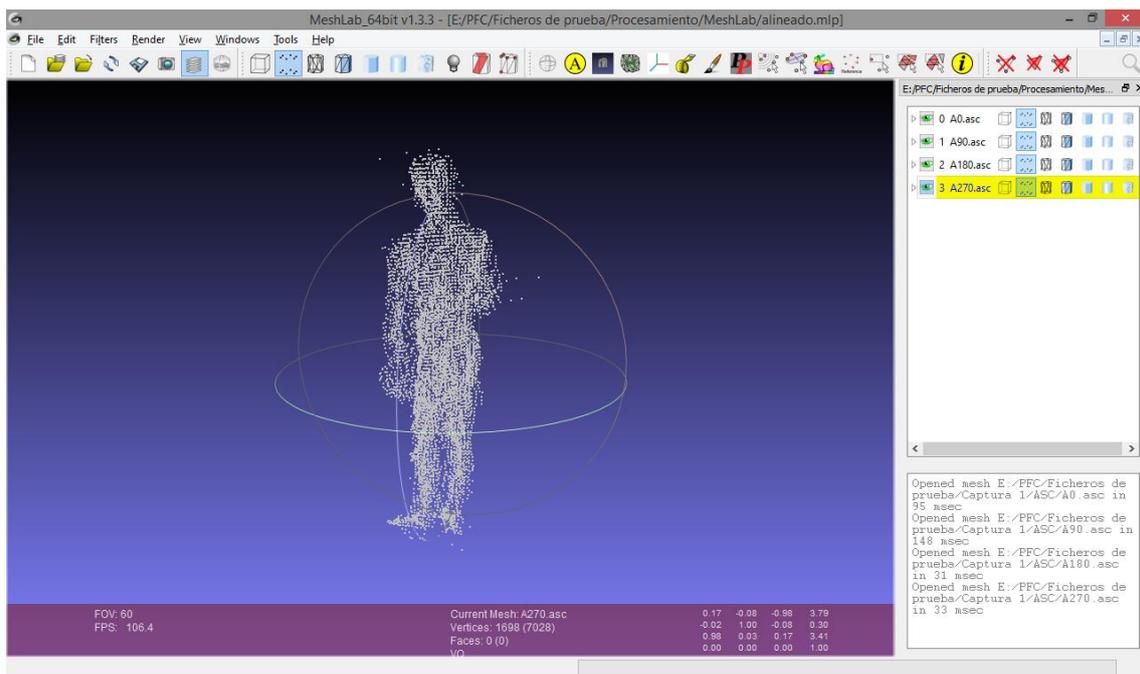


Ilustración 57. Estado final, perspectiva 2.

Podemos observar en los resultados como se ha conformado un volumen relativamente coherente si bien es cierto que aún requeriría un exhaustivo proceso de filtrado para eliminar redundancias, ruido superficial, etc.

5.3.2 Triangulación de un rostro

Una de las primeras pruebas realizadas fue la triangulación de un rostro. El proceso es similar a los descritos en el apartado 5.3, pero se ha decidido incluir esta prueba dado el nivel de detalle que se puede observar.

En este caso, se aplica un filtro de triangulación sobre la nube de puntos cuyo objetivo es producir una malla formada por triángulos que, como hemos visto con anterioridad, es una de las formas más comunes de representar un modelo tridimensional. Para ello, utilizamos el filtrado de Voronoi que nos proporciona el programa MeshLab, dado que no necesita información sobre las normales de cada vértice. Una vez que el filtro ha sido aplicado, la malla resultante se puede observar en la figura de la derecha.

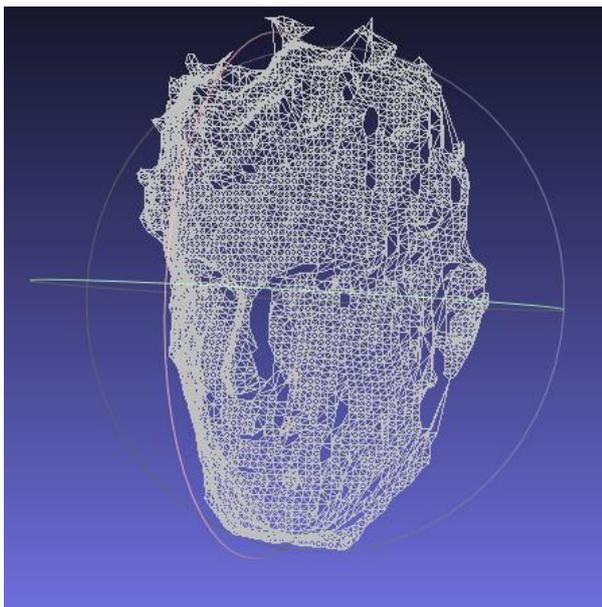


Ilustración 58. Triangulación de un rostro con MeshLab.

La siguiente ilustración representa la superficie obtenida de la triangulación del rostro. Como se puede observar, existen espacios vacíos en la representación tridimensional que se corresponden con falta de información (bien porque dada la perspectiva en la que se tomó la imagen no hay forma de escanear esa zona o bien porque debido a la poca confianza que ofrecían esos puntos fueron eliminados). Dada esa falta de información, el filtrado de Voronoi es incapaz de generar una triangulación válida en esa zona y por tanto no se generan las caras oportunas.

Por otra parte, con la triangulación realizada sobre la nube de puntos se puede observar con más detalle la precisión de captura de la cámara. A priori parece una buena representación pero

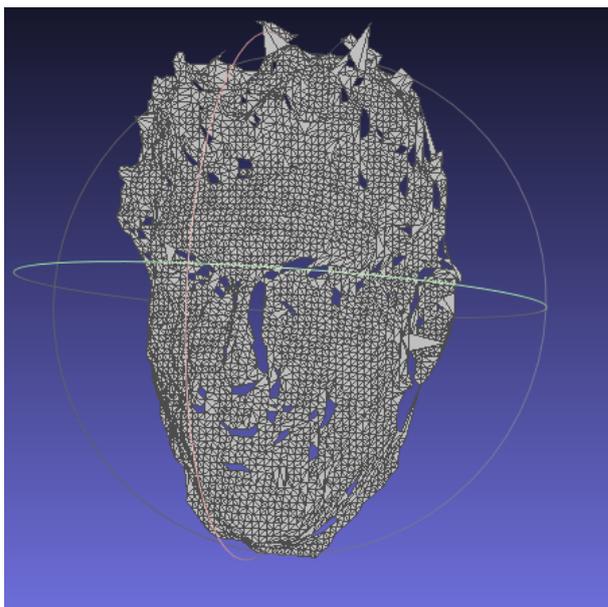


Ilustración 59. Superficie del rostro triangulado.

realizando diversos giros de cámara para observar con mayor detenimiento algunas zonas se puede comprobar como la representación está compuesta en su mayoría por superficies irregulares. Puede asemejarse al ruido que se genera en los datos tomados con cualquier sensor analógico. En cualquier caso, MeshLab proporciona diversos filtros cuya función es suavizar las superficies de los modelos tridimensionales, operación que en este caso puede ser de gran utilidad para corregir los defectos sobre la superficie del objeto tomado. Se trata fundamentalmente de filtros Laplacianos pero también permite utilizar otro tipo de

algoritmos. En cualquier caso, tras utilizar uno de los filtros Laplacianos existentes se puede observar como la superficie del modelo tridimensional se suaviza de forma que se produce un modelo más coherente.

5.4 Otros procesos de interés

En este apartado se describen otro tipo de procesos realizados o realizables sobre las nubes de puntos y que no guardan relación con los casos anteriores.

5.4.1 Suavizado mediante técnicas estadísticas básicas

Como se ha visto en las diferentes capturas realizadas con la cámara TOF SR4000, en mayor o menor medida se produce cierto ruido sobre la captura de una superficie, lo que provoca cierta inexactitud en los datos. Este problema es observable en la siguiente ilustración, que corresponde a una de las capturas realizadas sobre el torso del maniquí.

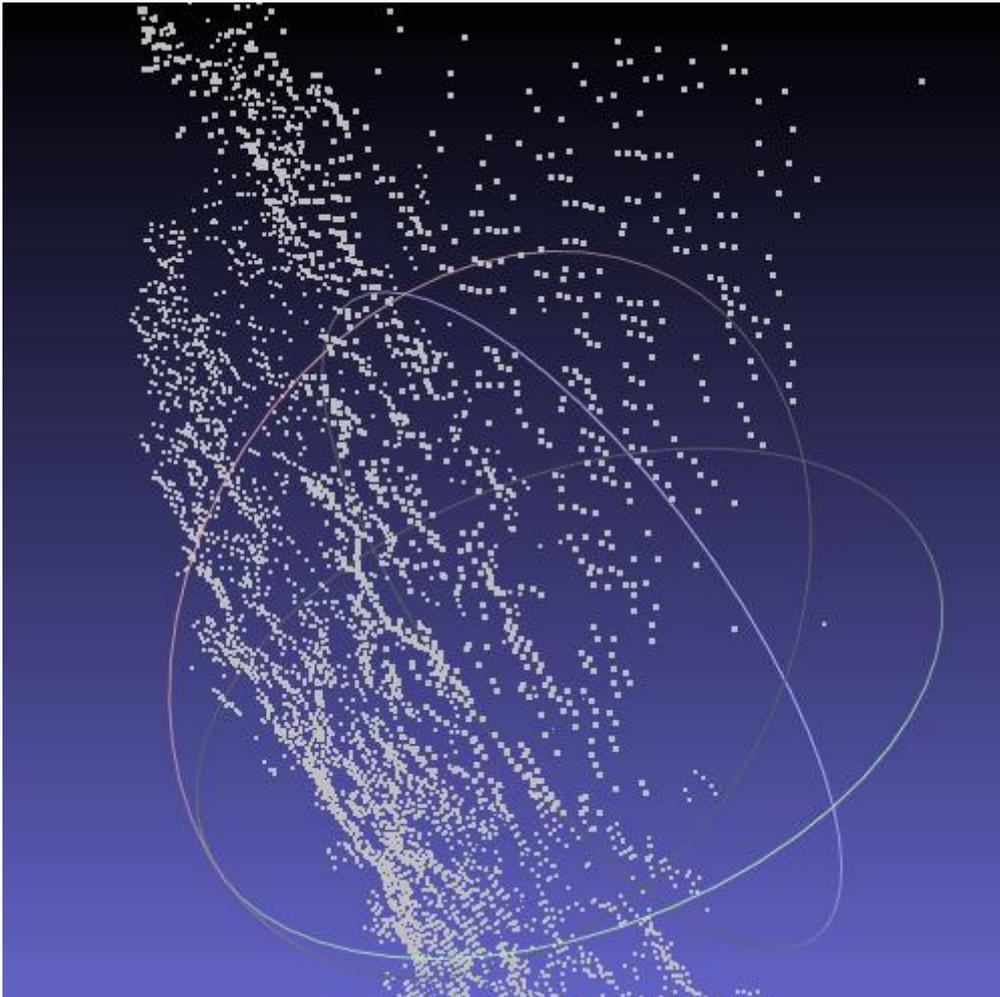


Ilustración 60. Fallo de captura de superficie.

Estas mismas pruebas repetidas sobre todas las capturas realizadas al maniquí nos arrojan el mismo resultado, un perfil correcto pero una superficie con ruido en la misma. A pesar de todo, los resultados no son extremadamente malos por lo que se hace viable la utilización de algún tipo de filtro para corregir la superficie.

En este caso, se aprovecha una de las capacidades de la cámara para corregir una de sus desventajas: la velocidad de captura. Dada la elevada tasa de frames por segundo que es capaz de capturar la cámara, se puede aplicar uno de los procesos que suelen utilizarse en procesos de medición como es la repetición de la toma de datos y aplicación de procesos estadísticos a dichas medidas. En este caso, se realizó una prueba simple para determinar si existe mejora

aplicando esta técnica. Sobre varios frames del objeto en la misma posición, podemos hacer una media de la posición de cada punto de forma que estadísticamente se reduzca la diferencia entre el valor medido y el valor real. Para ello, realizamos la prueba sobre el la primera captura con el maniquí tomando 10 frames y realizando la media sobre cada punto de cada uno de ellos. La captura utilizada es la primera de la secuencia mostrada en dicho caso, la que contiene el frontal del maniquí. El resultado, como podemos observar en la siguiente ilustración, es muy prometedor. El modelo de la derecha es el que ha sido sometido al suavizado utilizando la media de varios frames mientras que el de la izquierda es el original.

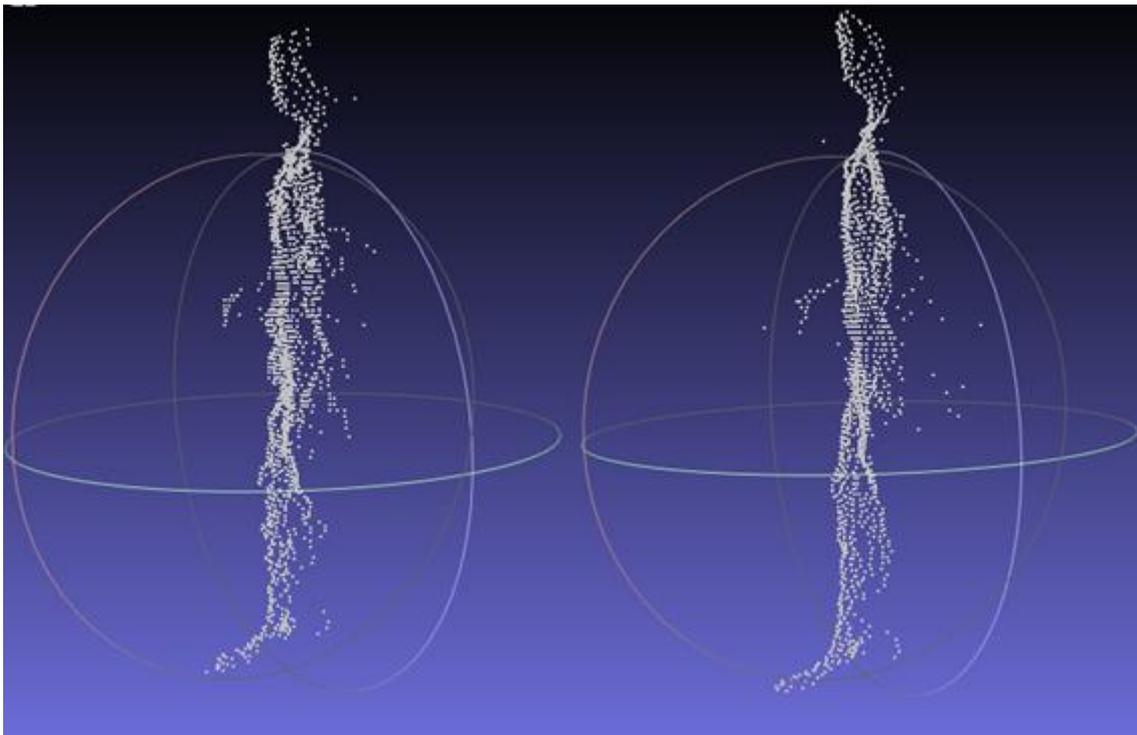


Ilustración 61. Suavizado utilizando varios frames.

Si bien es complicado apreciar el resultado en una imagen estática, si se puede comprobar como el modelo suavizado presenta un perfil más definido, con una superficie más lisa y coherente. Podemos deducir de este resultado que este tipo de suavizado es viable en la práctica si bien la teoría ya nos inducía a pensar en ese sentido. Es destacable el hecho de que se ha realizado de la forma más simple posible. Podría utilizarse un número mayor de frames y elementos estadísticos más avanzados.

Realizamos el mismo proceso sobre la segunda captura del maniquí y obtenemos el siguiente resultado. Al igual que en el caso anterior, el modelo de la derecha es el suavizado mientras que el de la izquierda es el original.

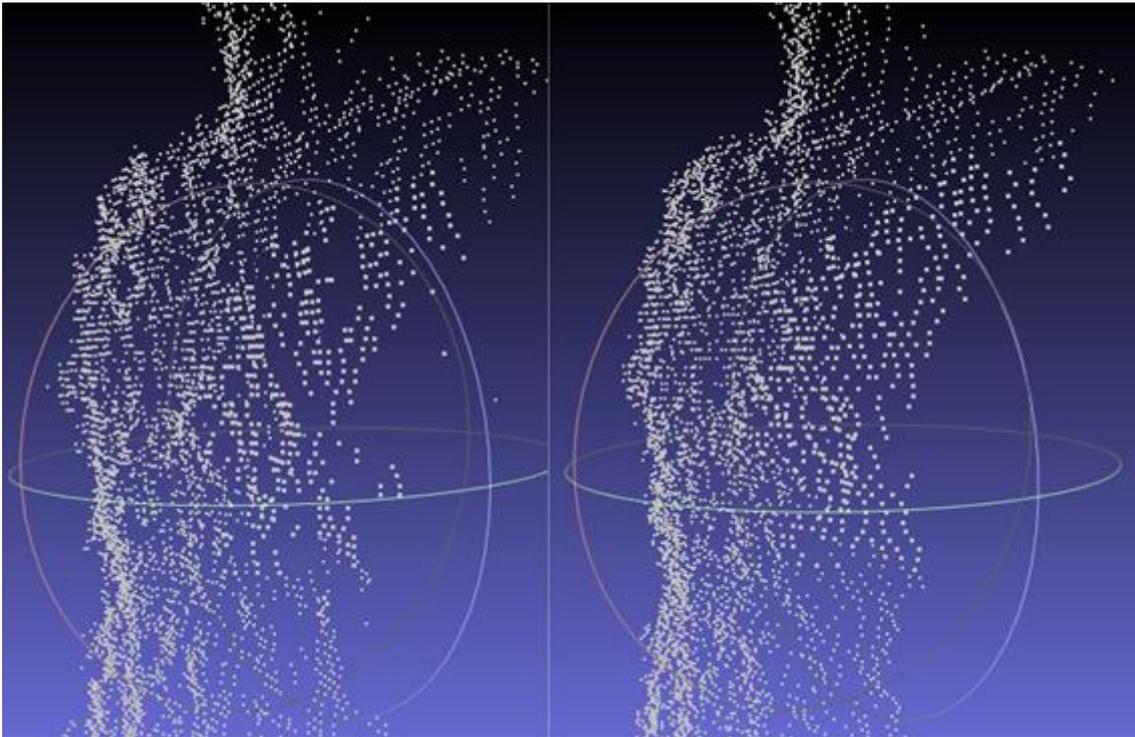


Ilustración 62. Segunda captura suavizada.

Si bien el resultado no mejora tanto el original como en el caso anterior, si se puede apreciar levemente una mejora en la definición de la superficie. Como conclusión a esta prueba podemos determinar que, si bien no en todos los casos produzca una mejora sustancial, el hecho de aplicar algún proceso estadístico sobre una serie de frames de un objeto puede mejorar la nube de puntos obtenida, más aún si los procesos aplicados son más avanzados que el utilizado para esta prueba.

5.4.2 Alineación automática de varias nubes de puntos

De la colaboración con José Manuel Maestre y Manuel Sillero surgió la idea de, partiendo de varias capturas realizadas sobre el mismo modelo, combinarlas de tal forma que formaran un volumen completo. Dicha combinación debería producirse utilizando una sola cámara, sin marcas de referencia y basándose únicamente en la rotación del modelo.

Como ya se ha observado durante las pruebas realizadas con MeshLab, es cierto que este objetivo podría cumplirse manualmente de forma fácil pero tediosa seleccionando puntos concordantes entre cada nube que queramos añadir o mediante la aplicación diversos valores de rotación y traslación hasta que se encontrara la alineación correcta. Sin embargo, el objetivo propuesto es realizarlo de forma automática, con la única ayuda quizá de indicar algún tipo de aproximación inicial en forma de rotación y traslación inicial pero sin llegar al nivel de realizar la operación prácticamente de forma manual.

Para conseguir el alineamiento existen varias técnicas, pero sin duda la más popular de todas es el algoritmo ICP descrito por Bels y McKay en *“A method for registration of 3-D shapes”* (10). Es un algoritmo iterativo que se caracteriza por el poco coste computacional en cada iteración, lo que se traduce en velocidad de cálculo. No obstante, presenta dos grandes inconvenientes: una imagen debe ser un subconjunto de la otra, ya que el algoritmo es muy sensible a la presencia

de outliers, y por otro lado es necesario conocer una buena aproximación inicial, porque de lo contrario el alineamiento podría ser defectuoso o podría requerir mucho tiempo en caso de que se probasen distintas posiciones iniciales. Este algoritmo está incluido tanto en MeshLab como en PCL y se ha probado su uso pero con resultados poco coherentes, producto probablemente del excesivo ruido superficial captado por la cámara. Es posible y está probado el uso de este algoritmo para realizar esta operación pero la obtención de un modelo coherente y lo más exacto posible podría constituir un proyecto en sí mismo, por lo que se optó por continuar con la visión más general de posibilidades ofrecidas por estas herramientas que por la obtención de un objetivo concreto.

CAPÍTULO 6. CONCLUSIONES

En este capítulo se recogen las conclusiones a las que se ha llegado tras finalizar el proyecto.

CAPÍTULO 6

6.1 LA CÁMARA SR400

6.2 SOFTWARE DESARROLLADO

6.3 MESHLAB

6.4 POINT CLOUDS LIBRARY (PCL)

6.5 ANTROPOMETRÍA

6.6 CONCLUSIONES FINALES

6.1 La cámara SR4000

Tras el uso continuado de la cámara durante las pruebas de laboratorio se ha podido llegar a las siguientes conclusiones que se cree deben tenerse en cuenta a la hora de trabajar con ella.

En primer lugar y como se mencionó anteriormente, en cuanto al manejo físico de la cámara hay que tener en cuenta que a pesar de parecer resistente debido a su fuerte carcasa, su sensor es extremadamente delicado. A su vez hay que tener cuidado con el filtro óptico que tenemos que evitar que se raye y mantenerlo limpio para obtener buenos resultados. Por otra parte, hay que extremar las precauciones al tocar la cámara en funcionamiento prolongado ya que su carcasa actúa a modo de disipador térmico y se calienta por encima de los 50º centígrados. El uso de un trípode adecuado para la cámara como el que se dispone en el laboratorio es altamente recomendable.

En segundo lugar, en cuanto a la precisión de la cámara, se ha podido comprobar durante las pruebas realizadas que los resultados son en general de gran calidad. Sin embargo, se ha detectado que la cámara comete fallos de medición importantes a la hora de capturar objetos con un alto nivel de brillo o materiales reflectantes. Es necesario controlar la iluminación de la escena dado que un exceso de iluminación puede interferir en una medición correcta. A su vez, los bordes de los objetos también suelen generar error aunque suele ser menos importante que el producido capturando superficies con alto nivel de brillo.

Por último, hay que destacar la gran velocidad de captura manteniendo gran precisión en las medidas llegando a superar los 25 fotogramas por segundo. Esto lo convierte en un sistema de captura de datos tridimensionales extremadamente veloz y permite su uso en campos que requieren gran velocidad de trabajo tales como la captura de gestos o el reconocimiento facial en base a la morfología de la cara.

6.2 Software desarrollado

En primer lugar, el software desarrollado para realizar estas pruebas, tanto el módulo de filtrado básico así como la conversión de ficheros para su manipulación con otros programas, funciona correctamente y es realmente útil para trabajar con las nubes de puntos producidas por la cámara SR4000. Se ha constatado que alguno de los filtros que no se han desarrollado pero si se han tenido en cuenta como posibles mejoras pueden ser de gran utilidad. En cualquier caso, el software ha cumplido correctamente su misión permitiendo tomar el elemento que se deseaba analizar separándolo del resto de datos superfluos para este estudio así como convertir dichos datos a los formatos de fichero apropiados para trabajar con cada herramienta.

Es importante mencionar que el API desarrollado por Daniel Sánchez en su proyecto ha sido de gran utilidad ya que ha permitido desarrollar este software de forma rápida y sencilla.

6.3 MeshLab

MeshLab es un software con una utilidad fuera de toda duda en cualquier proyecto relacionado con nubes de puntos o modelos tridimensionales. Uno de los puntos fuertes de este software es su gran cantidad de herramientas, tanto manuales como automáticas.

Uno de los puntos que hacen especialmente interesante esta herramienta es su gran versatilidad. Por un lado, puede usarse como programa al uso pero dada su estructura puede usarse como librería. Esto se consigue gracias a que cada una de las herramientas implementadas en MeshLab se encuentra en una librería independiente y el propio programa hace mención de que librería usa para cada herramienta. Esto permite realizar pruebas sobre el propio MeshLab para luego implementar la funcionalidad específica que queramos en nuestro propio proyecto reutilizando dichas librerías. La mayoría de ellas se regula bajo licencia GPL por lo que no habrá problemas de licencia pero es recomendable revisarlo en cada caso. Por otro lado, muchas de estas herramientas están basadas en algoritmos publicados en artículos científicos (conocidos comúnmente como papers) y en MeshLab se hace mención a dicho artículo por lo que podemos consultarlo para saber cómo funciona la herramienta.

Respecto al uso de MeshLab como librería en nuestros programas hay que mencionar que la documentación al respecto escasea o es difícil de encontrar por lo que no resulta fácil utilizar estas funciones fuera del entorno de MeshLab. Suele ser necesario investigar el código fuente de la aplicación para conocer el API de cada librería.

Respecto a la ventaja anterior, hay que mencionar que algunas de las herramientas contenidas presentan fallos de funcionamiento que provocan el cierre inesperado del programa. Esto puede deberse a que es difícil mantener funcionando correctamente una gran multitud de herramientas que puedan aplicarse en muy diversos casos. Sin embargo, gracias a que el programa se encuentra en continuo proceso de desarrollo y su modelo de código abierto en el que es posible contribuir a mejorar el código, cada nueva versión presenta menos fallos a la vez que incorpora nuevas herramientas y posibilidades.

Otro punto interesante del programa MeshLab es que está disponible en las plataformas más importantes hoy en día, incluidas plataformas móviles. Presenta idéntica funcionalidad en las plataformas de computadores más extendidas como son Windows, todas las distribuciones de Linux para PC y Mac OS. A su vez, su presencia en dispositivos móviles es muy interesante si bien aún no consigue toda la potencia disponible en las plataformas anteriormente citadas. Esto se debe a motivos obvios como pueden ser la baja capacidad de computación de estos dispositivos frente a una computadora o la dificultad de manejar un programa de esta magnitud en una pantalla pequeña y táctil.

Un punto negativo a destacar es la dificultad para encontrar documentación respecto al uso del programa. En la página web del proyecto (<http://meshlab.sourceforge.net>) apenas se hace mención de las diversas capacidades del programa y a los links de descarga del mismo. Cuenta con un foro de apoyo en el que se pueden consultar las dudas que surjan del trabajo con la herramienta pero no es el mejor sistema para iniciarse en el trabajo con el programa.

Otra de las características interesantes de MeshLab es que dado su desarrollo modular, podemos desarrollar nuestros propios plug-ins de forma que añadamos funcionalidad al

programa manteniendo todo lo ya existente. Sin embargo, como ya se mencionó anteriormente, la ausencia de documentación oficial al respecto dificulta la labor de crear una nueva funcionalidad.

6.4 Point Clouds Library (PCL)

La librería PCL es un proyecto extremadamente importante en el futuro del trabajo con nubes de puntos dada su intención de convertirse en el estándar en el tratamiento de este tipo de información. No solo cuenta con la intención de serlo sino que cuenta con el apoyo de grandes empresas tanto en su desarrollo como en su financiación.

El principal atractivo de la librería es su extensa lista de funcionalidades así como su potencia de procesamiento. Posee múltiples herramientas para trabajar con nubes de puntos y modelos tridimensionales así como algunas para trabajar con nubes bidimensionales. Incluye herramientas para trabajar directamente con diversos sensores por lo que se puede integrar con diverso hardware sin necesidad de herramientas intermedias. Esta desarrollada en C++ y suele utilizarse este lenguaje para desarrollar software que la utilice si bien es cierto que existen algunos wrappers que permiten utilizar la librería con otros lenguajes, pero no son demasiado potentes ni tienen mucho soporte.

Otros de los puntos fuertes de PCL es su continuo desarrollo con el que se consigue aumentar la funcionalidad, el rendimiento y reducir los errores de la librería. Este desarrollo no sería posible si PCL no hubiera sido concebido como un proyecto libre y abierto de forma que cientos de personas pueden colaborar en el desarrollo de la herramienta. Actualmente muchas empresas colaboran en el desarrollo así como infinidad de programadores libres que aportan sus conocimientos y su tiempo al desarrollo de la librería y al soporte de la misma. Esto permite la continuidad citada, cosa que es muy importante pues muchos proyectos, a pesar de ser interesantes y útiles, mueren al carecer de una comunidad extensa que les de soporte.

Por otro lado, PCL cuenta con una amplia documentación en su página web (<http://www.pointclouds.org/documentation/>) así como múltiples formas de contacto a través de las cuales solicitar ayuda para resolver los problemas que puedan surgir durante el desarrollo de aplicaciones con PCL. Existen dos listas de correo fundamentales (<http://www.pcl-users.org> y <http://www.pcl-developers.org>) para usuarios y desarrolladores en las que hay muchísima información respecto a problemas ya solucionados y en las que se puede realizar consultas sobre el uso de la librería. Son listas de correo muy activas por lo que las respuestas a las dudas no suelen tardar en llegar.

Respecto a la documentación, hay que destacar que en la propia web podemos encontrar diversos tutoriales que nos ayudaran a manejar los conceptos básicos de cada módulo, incluyendo código de ejemplo y diversas posibilidades de compilación del mismo.

Otro punto fundamental de PCL es la modularidad y el hecho de que se pueda compilar para múltiples plataformas. Gracias a esto puede implementarse en múltiples dispositivos incluso en aquellos de bajas especificaciones técnicas utilizando únicamente los módulos necesarios para nuestra aplicación.

Por todo lo mencionado, se considera que PCL es una herramienta extraordinaria a la hora de realizar proyectos de software relacionados con nubes de puntos gracias a la cual se simplificará el trabajo a realizar con dichos datos, permitiendo aprovechar el tiempo en el desarrollo del concepto de la aplicación en lugar de tener que programar las bases matemáticas y algoritmos necesarios para trabajar con las nubes de puntos. En este aspecto, entra en juego la llamada

reutilización, término que en informática se aplica al uso de herramientas o código ya hecho y que se conoce que funciona correctamente para agilizar los desarrollos.

6.5 Antropometría

Dada la colaboración en la realización de las pruebas, se considera oportuno hacer una pequeña mención de las conclusiones a las que ha llegado José Manuel Maestre durante el estudio.

Por un lado, los resultados no han sido todo lo buenos que se esperaba pero si son alentadores de cara al futuro uso de la tecnología TOF en antropometría. Estos resultados obviamente están condicionados por el dispositivo de captura por lo que una mejora en el mismo arrojaría mejores resultados.

Respecto a las ventajas del uso de la cámara SR4000, José Manuel menciona que supone una forma más económica y con más disponibilidad de transporte que los sistemas utilizados hasta ahora. Se determina que la precisión de la cámara a la hora de realizar medidas es lo suficientemente alta dado que la diferencia de medida respecto a técnicas antropométricas tradicionales no es estadísticamente significativa. Por otra parte, la velocidad de adquisición de datos es muy superior a otros sistemas de escaneado.

Respecto a los inconvenientes observados, tal y como se ha mencionado ya en este estudio, la cámara tiene dificultades para capturar objetos cuya superficie posea un brillo excesivo. Por otra parte, se constata que otro de los problemas que se pueden encontrar a la hora de capturar un cuerpo humano es el de las zonas ocultas. Para evitarlo, se recomienda la aplicación de la norma ISO 20685:2010 (17) que establece las pautas a la hora de utilizar un escáner tridimensional.

Por otra parte, destaca el hecho de que no existe demasiado software específico para esta tecnología y este campo. Existe determinado software, como MeshLab, que puede ser utilizado para realizar un estudio antropométrico básico pero al no ser esa su función se queda generalmente corto de funcionalidad para temas más avanzados. Por tanto, parece ser necesaria la investigación y desarrollo de software para este campo.

Para finalizar, durante el estudio se intentó realizar una medición volumétrica de un cuerpo humano utilizando una sola cámara y sin marcadores mediante rotación del elemento a capturar. Debido a la complejidad del proceso que bien podría afrontarse como un proyecto propio, no se lograron resultados relevantes durante las pruebas. Sin embargo si se sentaron las bases localizando diversas técnicas y herramientas aplicables a este proceso. También se menciona la posibilidad de utilizar varias cámaras de forma simultánea, cosa que facilitaría el trabajo pero aumentaría los costes.

6.5 Conclusiones finales

Para finalizar, teniendo en cuenta la visión global del proyecto podemos extraer las siguientes conclusiones.

En primer lugar, el software desarrollado ha sido muy importante así como la base de la que parte (el API de Daniel Sánchez). Es cierto que este software es dependiente del modelo de cámara utilizado y que variando este elemento el software podría dejar de tener utilidad, pero centrándonos en los medios con los que se contaba en el proyecto ha sido indispensable. En primer lugar como herramienta para realizar un primer filtrado con los datos en bruto su utilidad está fuera de dudas, pues prácticamente en segundos podemos obtener una nube de puntos filtrada. Por otra parte, la generación de ficheros con los que trabajar en MeshLab y en PCL ha sido fundamental. Existe una herramienta de demostración de la cámara SR4000 capaz de exportar nubes de puntos en formato ASC utilizables por MeshLab pero sus capacidades de filtrado y trabajo frame a frame son bastante limitadas. Solo supera al software desarrollado en que es capaz de mostrar directamente la nube de puntos resultante de forma tridimensional.

Respecto a MeshLab y PCL, destacar que ambas son grandes herramientas para trabajar con nubes de puntos y datos tridimensionales. Sin embargo cada una de ellas puede aplicarse en un ámbito distinto. Por un lado, MeshLab es superior como herramienta de análisis inicial dado que posee una interfaz gráfica que hace su uso extremadamente sencillo. Una vez obtenida una nube de puntos es muy rápido visualizarla y trabajar con ella en MeshLab. Por el contrario, PCL es muy superior como librería a usar en el desarrollo de nuestro propio software específico. Si bien es cierto que MeshLab también puede ser utilizado de esta forma, PCL en su concepto está orientado a ser una librería estándar en el procesamiento de nubes de puntos, está ampliamente documentada y posee una comunidad muy activa tanto en su desarrollo como en su soporte.

Respecto al ámbito antropométrico, como comenta José Manuel, las pruebas quizá no hayan sido excesivamente buenas pues la cámara quizá no alcance la precisión requerida pero si indican la viabilidad de esta tecnología. De hecho, los resultados de las pruebas están condicionados al dispositivo de captura utilizado por lo que es posible que con otro dispositivo con tecnología TOF se consigan mejores resultados. Respecto al software, parece evidente que para un campo tan específico es necesario el uso o desarrollo de software destinado exclusivamente a la antropometría. Por otra parte, el hecho de intentar generar un modelo tridimensional completo utilizando una única cámara y sin marcas de referencias si bien no ha generado un resultado satisfactorio si puede sentar las bases de un futuro trabajo.

Para finalizar, como línea futura de trabajo y manteniendo la antropometría como marco práctico, se pueden tomar dos vías de trabajo:

- Generación de un modelo tridimensional completo sin marcas de referencia utilizando únicamente una cámara TOF mediante rotación del modelo.
- Generación de un modelo tridimensional completo sin marcas de referencia integrando las capturas de varias cámaras TOF de forma simultánea.

Ambos proyectos parecen a priori interesantes y con una aplicación práctica muy marcada.

ANEXOS

En estas páginas se encuentran los anexos al documento.

ANEXOS ANEXO I. PRESUPUESTO DEL PROYECTO

ANEXO II. LA CÁMARA SR400

ANEXO III. INTRODUCCIÓN A LA ANTROPOMETRÍA

ANEXO IV. UN EJEMPLO PRÁCTICO DE ESCANEADO TRIDIMENSIONAL

ANEXO V. CÓDIGO FUENTE DE LAS PRUEBAS

ANEXO I. Presupuesto del proyecto

En este anexo se detalla el presupuesto que se ha calculado necesario para llevar a cabo este proyecto.

Costes

En las siguientes tablas se describen los diferentes elementos que han intervenido en el desarrollo de este proyecto y se especifica su precio total.

Precio del Hardware

ELEMENTO	DESCRIPCIÓN	PRECIO
PC	Ordenador personal para realizar el trabajo. Se incluyen todos los elementos Hardware requeridos tales como monitor, teclado o ratón.	800,00 €
Cámara MESA Imaging SR4000	Cámara con tecnología TOF utilizada para realizar las pruebas. Esta cámara incluye todos los elementos necesarios para su funcionamiento incluido el cableado necesario.	3.500,00 €
Trípode	Trípode compatible con la cámara SR4000 utilizado para mejorar la estabilidad de la misma a la hora de realizar las pruebas.	35,00 €

Tabla 9. Precio hardware del presente proyecto.

Precio del Software

ELEMENTO	DESCRIPCIÓN	PRECIO
Microsoft Windows 7 Professional 64 bits	Sistema operativo para el ordenador personal utilizado para realizar el proyecto.	125,00 €
Microsoft Office 2010 Hogar y Estudiantes	Suite de ofimática utilizada para realizar la documentación del proyecto.	125,00 €
Microsoft Visual Studio 2010 Professional	Entorno de desarrollo integrado utilizado para desarrollar todo el software que ha sido necesario programar para este proyecto.	800,00 €

Tabla 10. Precio software del presente proyecto.

Precio del personal

ELEMENTO	DESCRIPCIÓN	PRECIO
Ingeniero en Informática	Persona requerida para realizar el trabajo de investigación y diseño para este proyecto.	20,00 € /hora
Programador	Persona encargada de desarrollar el código fuente del software y pruebas realizadas para este proyecto.	15,00 € /hora

Tabla 11. Precio del personal del presente proyecto.

Presupuesto final

En la siguiente página se especifica el presupuesto completo del proyecto tomando el precio de cada elemento que ha intervenido en el proyecto de las tablas anteriores. Respecto a dicho presupuesto, es necesario mencionar las siguientes consideraciones:

- Se ha tenido en cuenta que los elementos hardware son amortizables a lo largo del tiempo. Se ha considerado como ciclo de vida útil un periodo de cinco años o lo que es lo mismo, 60 meses. Por norma general esto no quiere decir que el dispositivo deje de funcionar al finalizar dicho periodo pero dado el avance actual de la tecnología, pasaría a considerarse obsoleto.
- De igual forma, se ha tenido en cuenta que los elementos software también son amortizables. En este caso, el ciclo de vida lo ha determinado el periodo de soporte especificado por el desarrollador. Dado que el software utilizado es desarrollado por Microsoft, dicho periodo ha sido consultado en su web de soporte (18).
- Respecto al coste del personal, si bien todo el trabajo ha sido realizado por una única persona, se han desglosado los costes en función de la tarea a realizar.
- El cálculo de horas por mes que se imputa en el coste del personal viene dado de estimar una media de veintidós días trabajados al mes a razón de ocho horas diarias.

PRESUPUESTO DEL PROYECTO

DESGLOSE DEL COSTE MATERIAL				
ELEMENTO	PRECIO	AMORTIZACIÓN	DEDICACIÓN	COSTE
PC	800,00 €	60 meses	6 meses	80,00 €
Cámara SR400	3.500,00 €	60 meses	6 meses	350,00 €
Trípode	35,00 €	60 meses	6 meses	3,50 €
Microsoft Windows 7 Professional 64 bits	125,00 €	132 meses	6 meses	5,70 €
Microsoft Office 2010 Hogar y Estudiantes	125,00 €	132 meses	6 meses	5,70 €
Microsoft Visual Studio 2010 Professional	800,00 €	132 meses	6 meses	36,40 €
Material fungible	100,00 €	N/D	N/D	100,00 €
PRECIO TOTAL DEL MATERIAL				5485,00 €
COSTE TOTAL IMPUTABLE AL PROYECTO				581,30 €

DESGLOSE DEL COSTE DE PERSONAL				
ELEMENTO	PRECIO	HORAS/MES	MESES	COSTE
Ingeniero en Informática	20,00 € / h	176 horas	2 meses	7.040,00 €
Programador	15,00 € /h	176 horas	4 meses	10.560,00 €
COSTE TOTAL DEL PERSONAL				17.600,00 €

COSTE TOTAL DEL PROYECTO	
ELEMENTO	COSTE
Coste total del material imputable al proyecto	581,30 €
Coste total del personal	17.600,00 €
COSTE FINAL	18.181,30 €

ANEXO II. La cámara SR4000

La cámara que se ha utilizado para realizar las pruebas de este proyecto es el modelo Swiss Ranger de la compañía MESA Imaging, en concreto el modelo SR4000. Esta cámara es la cuarta generación de cámaras tridimensionales con tecnología TOF de la compañía MESA Imaging. Como se puede apreciar en la imagen, la cámara es compacta y de pequeño tamaño (65 x 65 x 76 mm), pesando 510 gramos según especificaciones del fabricante. Dado el pequeño tamaño del aparato podría considerarse su uso en sistemas móviles portados por humanos pero su elevado peso en relación a su tamaño la hacen incomoda de transportar. Sin embargo, es idónea para acoplar en elementos mecánicos como pueden ser vehículos o robots. Su carcasa es resistente pero no así la lente por lo que debe estar adecuadamente protegida en entornos en los que puedan producirse daños al aparato. Por otra parte, se ha constatado en las pruebas que la cámara produce un calentamiento excesivo sobre su carcasa que actúa a modo de disipador por lo que hay que extremar las precauciones al manipularla tras su uso.



Ilustración 63. Cámara TOF SR4000.

Si bien existen versiones que únicamente aceptan conexión USB, la cámara utilizada en las pruebas de laboratorio puede ser conectada a la computadora mediante USB o Ethernet siendo este último método muy útil si quieren utilizarse varias cámaras de forma simultánea. Es necesario en todos los casos contar con un driver instalado que gestiona la conexión con la cámara y nos ofrece el API para trabajar con ella, además de encargarse del procesamiento de ciertos datos generados por la cámara.

Una de las grandes ventajas de esta cámara es su elevada precisión y frecuencia de muestreo, sin embargo, su elevado precio que ronda los 4.000 dólares no la hace asequible para todos los proyectos.

La tecnología TOF en el modelo SR4000

En este apartado se va a realizar una breve explicación de cómo funciona la tecnología TOF en este aparato.

Como se ha mencionado en capítulos anteriores, los sistemas de adquisición de imágenes tridimensionales basados en tecnología TOF se clasifican como activos sin contacto. Esto quiere decir que la cámara debe emitir algún tipo de radiación y a su vez debe registrar como esa radiación incide en la escena que se pretende capturar. En el caso de la cámara SR4000, se utiliza luz infrarroja emitida por 24 leds en el rango de 850 nm por lo que no es visible para el ser

humano y tampoco es dañino para los ojos si se mira directamente a la cámara.

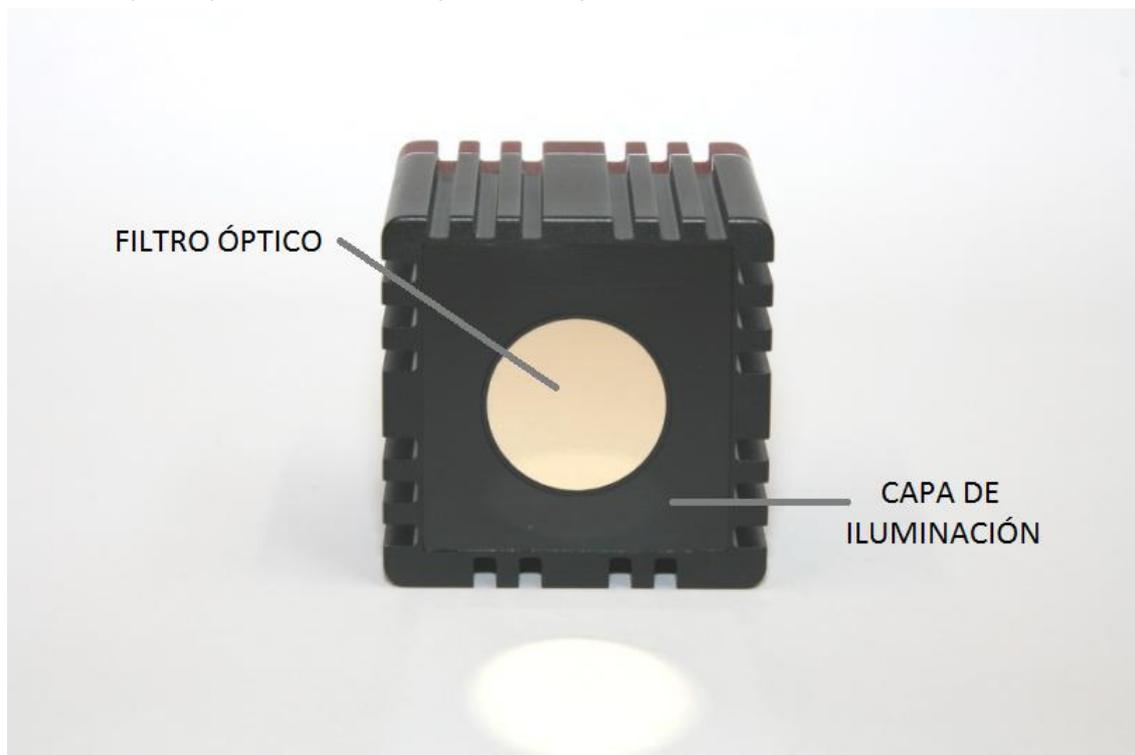


Ilustración 64. Frontal de la cámara SR4000

Como puede observarse en la figura anterior, el emisor está conformado por una cobertura, denominada en la imagen capa de iluminación, que protege los 24 leds emisores permitiendo que la luz emitida por estos sea transmitida al exterior. Por su parte, el receptor está formado en su parte visible por el filtro óptico cuya función es filtrar la luz que llega al sensor permitiendo solo pasar la comprendida en el rango cercano a la emisión de los leds.

Para determinar la distancia en cada pixel, la cámara determina el tiempo que tarda la luz en llegar a los objetos de la escena y volver. Dado que la velocidad de la luz es conocida y constante, podemos determinar la distancia directamente con este tiempo medido. Para calcular el tiempo, la cámara modula los leds de iluminación y el sensor mide la fase de la señal modulada retornada en cada pixel. La distancia es calculada con la siguiente fórmula:

$$D = \frac{c}{2f}$$

La constante “c” es la velocidad de la luz y la constante “f” es la frecuencia de modulación. Dado que la cámara funciona generalmente con una frecuencia de modulación de 30 Mhz, la distancia máxima a la que la cámara es capaz de tomar medidas precisas es de 5,00 metros. Este valor de modulación puede ser configurado resultando en pequeñas variaciones en la distancia máxima. En cualquier caso, este dato teórico ha sido corroborado durante las pruebas de laboratorio.

Modos de captura y resultados

El modelo SR4000 tiene dos modos de captura con los que se puede trabajar:

- **Adquisición continua.** En este modo la cámara trabaja como si de una cámara de video se tratase capturando datos de forma continua. En este modo, la cámara capturará una secuencia de fotogramas. Durante las pruebas de laboratorio la cámara ha mostrado una gran velocidad de captura superando los 25 fotogramas por segundo en función de algunos parámetros de ajuste en la captura.
- **Adquisición disparada.** En este caso, la cámara se comporta como si de una cámara de fotos se tratase. Puede utilizarse captura mediante orden hardware o software. En cualquiera de los casos, la cámara procesará un único fotograma.

En cualquiera de los dos modos de adquisición, la cámara genera fotogramas de 176 por 144 píxeles de tamaño. Por cada píxel obtenemos tres valores que indican amplitud capturada, distancia y confianza en la medida. A continuación se describen los diferentes datos que se obtienen por cada captura y se incluye una imagen de muestra que representa gráficamente los mismos. Hay que tener en cuenta que las imágenes están ampliadas a 352 por 288 píxeles dado que en su tamaño original son demasiado pequeñas para observarlas fácilmente.

El primero y más importante de los datos que nos retorna la cámara tras cada captura es la distancia correspondiente a cada píxel de la escena. En la ilustración mostrada a continuación, podemos observar una representación de los datos de distancia tomados en una captura. Cada píxel puede tomar un valor de 16 bits entre 0x0000 que se correspondería con distancia 0, es decir, en el frontal mismo de la cámara, y 0xFFFF que se correspondería, en un sistema radial, con un retorno de señal con fase completa o lo que es lo mismo, una distancia de 5,00 metros en el caso de utilizar una modulación de 30 Mhz. Este valor es considerado como el dato en bruto de la distancia. En conjunto con este valor, la cámara nos devuelve también este dato procesado a través del driver de la computadora de forma que nos ofrece la distancia de cada píxel en un sistema de coordenadas cartesianas expresado en metros. Este es el valor resultado más útil de todos ya que para cada píxel nos ofrece el vector (x, y, z) con el que podemos generar una nube de puntos de la escena capturada.

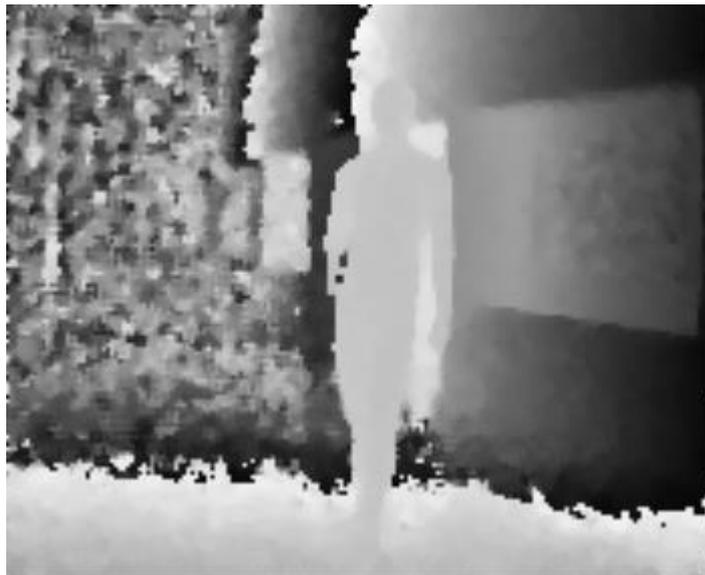


Ilustración 65. Imágen de distancia

El segundo de los datos es la amplitud. Este valor es el que utiliza la cámara para realizar el cálculo de distancia y también es un valor de 16 bits. Dado que la cámara ya nos ofrece el cálculo de la distancia en cada píxel, este dato puede no resultar de gran utilidad en la mayoría de los casos. Sin embargo, puede utilizarse de forma general como una imagen en escala de grises de la escena capturada. En la siguiente ilustración podemos observar la representación de la amplitud capturada en la misma escena que la ilustración anterior. Se puede ver que es posible utilizarla como imagen en escala de grises como referencia del fotograma que se está estudiando a pesar de que la calidad de la imagen no es excesivamente buena.



Ilustración 66. Imagen de amplitud

Por último, a través del driver instalado en la computadora, la cámara nos ofrece un “mapa de confianza”. Este mapa nos indica para cada píxel un porcentaje de verosimilitud de la medida tomada. Este valor es un entero de 16 bits representando a mayor valor, mayor verosimilitud. Es generado usando una combinación de las medidas de distancia y amplitud y de su variación en el tiempo. Es muy útil para analizar una captura dado que nos permite descartar píxeles de baja verosimilitud o seleccionar únicamente aquellos que son altamente confiables. A continuación podemos ver una representación gráfica del mapa de confianza en el que las zonas más oscuras representan menor confianza y las zonas más claras mayor confianza.



Ilustración 67. Imagen de confianza

ANEXO III. Introducción a la antropometría

En este capítulo se realiza una pequeña descripción del campo de la antropometría, para el cual se estudian diversas alternativas que permiten facilitar el trabajo mediante medios informatizados y para el cual se estudia la viabilidad de la tecnología TOF de la cámara SR4000.

Definición de antropometría

La palabra antropometría significa “la medida del hombre” y es una ciencia que constituye una rama de la antropología biológica o física cuyo objetivo es el estudio de las medidas del hombre.

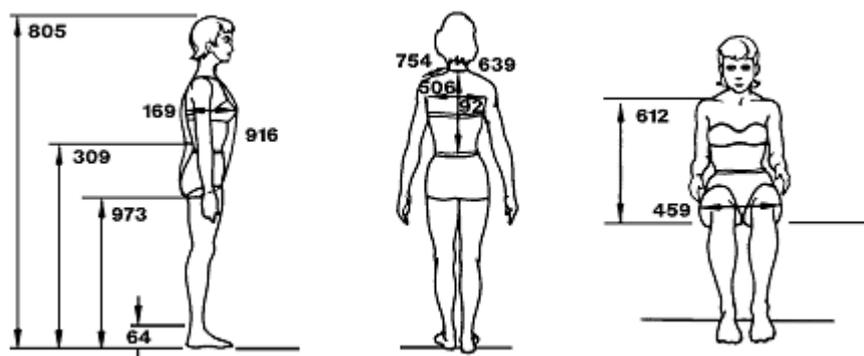


Ilustración 68. Ejemplo de antropometría.

Una de las características de la antropometría es que el resultado de la toma de medidas debe ser exacto, es decir, sin ningún tipo de margen de error. Entre otros, el objetivo primordial que concibe la antropometría es el estudio de las dimensiones y medidas humanas con el propósito de evaluar los cambios físicos del hombre así como diferenciar entre sus diferentes razas y subrazas. Es interesante recalcar el hecho de que no permite margen de error dado que esto será muy importante a la hora de analizar los resultados obtenidos de las pruebas de concepto.

Esta ciencia encuentra su origen en el siglo XVIII en el desarrollo de estudios de antropometría racial comparativa por parte de antropólogos físicos. Sin embargo, no fue hasta 1870 con la publicación de "Antropometrie", del matemático belga Quételet, cuando se considera su descubrimiento y estructuración científica. Pero fue a partir de 1940, con la necesidad de datos antropométricos en la industria, específicamente la bélica y la aeronáutica, cuando la antropometría se consolida y desarrolla, debido al contexto bélico mundial.

Las dimensiones del cuerpo humano varían de acuerdo al sexo, edad, raza, nivel socioeconómico, etc.; por lo que esta ciencia dedicada a investigar, recopilar y analizar estos datos, resulta una directriz en el diseño de los objetos y espacios arquitectónicos, al ser estos contenedores o prolongaciones del cuerpo y que por lo tanto, deben estar determinados por sus dimensiones. Estas dimensiones son de dos tipos importantes: estructurales y funcionales. Las estructurales son las de la cabeza, troncos y extremidades en posiciones estándar. Mientras que las funcionales o dinámicas incluyen medidas tomadas durante el movimiento realizado por el cuerpo en actividades específicas. Al conocer estos datos se conocen los espacios mínimos que el hombre necesita para desenvolverse diariamente, los cuales deben de ser considerados en el diseño de su entorno. Aunque los estudios antropométricos resultan un importante apoyo para saber la relación de las dimensiones del hombre y el espacio que éste necesita para realizar

sus actividades, en la práctica se deberán tomar en cuenta las características específicas de cada situación, debido a la diversidad antes mencionada logrando así la optimización en el proyecto a desarrollar.

En el presente, la antropometría cumple una función importante en el diseño industrial, en la industria de diseños de indumentaria, en la ergonomía, la biomecánica y en la arquitectura, donde se emplean datos estadísticos sobre la distribución de medidas corporales de la población para optimizar los productos.

Referencias anatómicas

A la hora de realizar un estudio antropométrico es necesario realizar las conocidas como referencias anatómicas. Estas no son más que puntos que se marcan en la piel, los cuales sirven para identificar la ubicación exacta del sitio que se tomará como referencia para efectuar la medición. Generalmente corresponden a la proyección cutánea de referencias óseas.

Tradicionalmente, estos puntos se ubican a través del tacto, procurando evitar el desplazamiento de la piel sobre la referencia ósea para marcar posteriormente con un lápiz dermatográfico, realizándose todas las marcas antes de comenzar con la medición.

Las referencias anatómicas más tenidas en cuenta como norma general son las que se detallan a continuación.

- **Acromial.** Punto ubicado en la parte más superior y lateral del proceso acromial.
- **Radial.** Punto ubicado en el borde más proximal (o superior) y lateral de la cabeza del radio.
- **Punto medio acromio-radial.** Punto equidistante entre las marcas acromial y radial. Se debe trazar una línea horizontal hasta las caras anterior y posterior del brazo para luego poder ubicar los sitios de medición de los pliegues bicipital y tricipital.
- **Estiloideo.** Es el punto más inferior (o distal) de la apófisis estiloidea radial.
- **Estiloideo medio.** Es el punto medio, en la superficie anterior de la muñeca, sobre una línea horizontal que pasa a nivel del punto estiloideo.
- **Subescapular.** Se ubica primero el ángulo inferior del omóplato y desde allí se traza una diagonal hacia abajo y hacia la derecha de 2 cm. donde finalmente se marca el punto subescapular. Esta marca se la ubica de ésta manera debido a que los pliegues nunca se toman sobre zonas óseas subyacentes.
- **Meseoesternal.** Punto que se marca en la línea media, sobre el esternón, a la altura de la mitad de la articulación de la 4^o costilla.
- **Xifoideo.** Punto que se ubica en la extremidad inferior del esternón, inferior a la apófisis xifoidea.
- **Axilar.** Punto que se marca en la intersección de una proyección horizontal imaginaria del punto esternal con la LÍNEA ILIOAXILAR que es una línea vertical imaginaria que pasa por el centro del hueco axilar. Se debe mantener el brazo horizontal en abducción.
- **Iliocrestídeo.** Es el punto más lateral de la cresta iliaca sobre la línea ilioaxilar.

- **Ilioespinal.** Se marca primero la espina iliaca anterosuperior, luego se marca el punto ilioespinal en la intersección de una línea imaginaria que une la espina iliaca anterosuperior y el ángulo de la axila con otra línea imaginaria que es la proyección horizontal del punto iliocrestídeo.
- **Trocantéreo.** Es el punto más superior del trocánter mayor del fémur.
- **Tibia medial.** Punto más superior del borde medial de la meseta tibial.
- **Maleolar.** Es el punto más distal o inferior del maléolo medial o tibial.
- **Tibial lateral.** Es el punto más superior del borde lateral de la meseta tibial.

Es necesario recalcar que se han descrito los puntos de referencia de la antropometría general. En casos más específicos, como puede ser la antropometría facial, es necesario determinar otra serie de puntos más apropiados para el estudio.

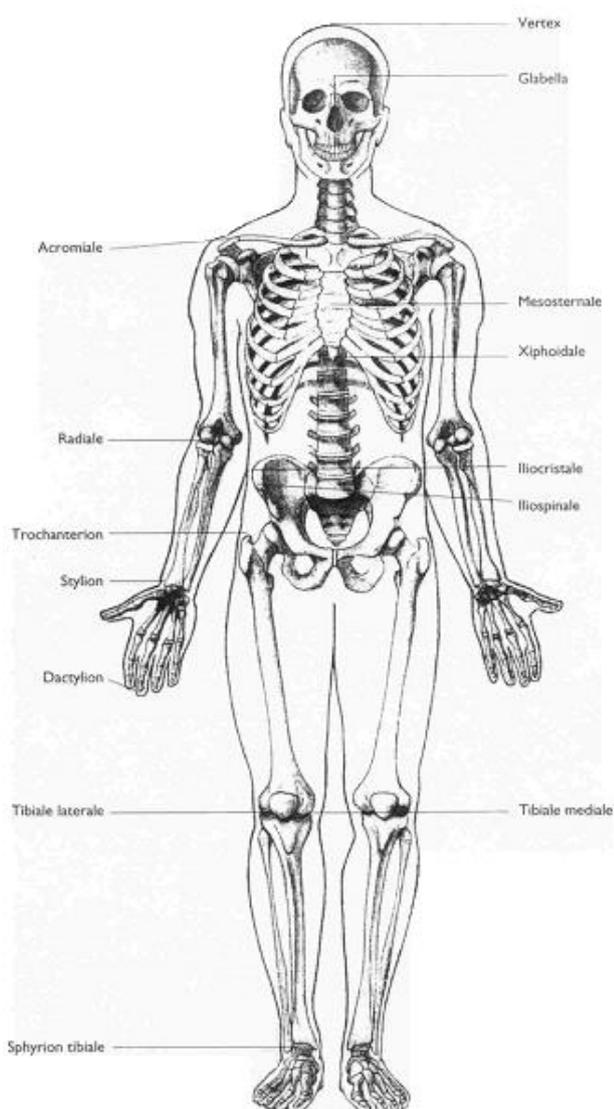


Ilustración 69. Referencias antropométricas generales.

La técnica antropométrica tradicional

Dado el objetivo de esta ciencia, es evidente que tradicionalmente se han utilizado diversos aparatos manuales para realizar las mediciones oportunas sobre el cuerpo humano. Generalmente, los más usados son los siguientes:

- **Cinta antropométrica.** La toma de perímetros se realiza con una cinta inextensible, preferentemente de acero flexible, calibrada, no más ancha de 7 mm y un espacio en blanco de al menos 3 cm antes de la línea de registro del 0. Además de medir perímetros se la utiliza para ubicar en forma precisa distintos sitios de pliegues cutáneos y para marcar distancias desde los puntos o referencias anatómicas.
- **Estadiómetros.** Es una regla calibrada ubicada verticalmente, generalmente adosada a la pared. Tiene una pieza deslizante que se baja hasta el vértex de la cabeza. Se lo usa para medir la estatura y la altura sentado.
- **Balanzas.** Se utilizan balanzas de pesas con precisión cercana a los 100 gr.

- **Plicómetros.** Son calibres para la medición de pliegues cutáneos. Se sugieren calibres que realicen una compresión de 10gr/mm².
- **Antropómetro.** Son instrumentos que se utilizan para medir las alturas verticales entre puntos o referencias anatómicas específicas del sujeto y el piso.

- **Segmómetro.** Es una barra de acero calibrada la cual posee en su extremo una rama recta, fija, de aproximadamente 7 cm de longitud, y otra deslizante del mismo largo. Es un instrumento que sirve para medir longitudes segmentarias directamente (acromio-



Ilustración 70. Fotografía de un segmómetro.

- radial, radial-estiloidea) y algunas alturas (trocanterea, ilioespinal). Está diseñado para ser utilizado en reemplazo del antropómetro.
- **Calibres deslizantes grandes.** Es parecido al segmómetro, tiene ramas largas y se lo utiliza para la medición de grandes diámetros óseos por ejemplo: biiliocrestideo o biacromial.
- **Calibres deslizantes pequeños.** Son calibres pequeños de ramas cortas. Se los utiliza para la medición de pequeños diámetros óseos como los biepicondilares del húmero y del fémur, biestiloideos y bimaleolares.
- **Calibres de ramas curvas.** Sus ramas son curvas lo cual permite la medición del diámetro antero – posterior del tórax.
- **Caja antropométrica.** Es una caja cúbica que mide 40 cm en todos los lados de sus caras. Se utiliza como una ayuda para medir alturas. Por ejemplo para tomar la altura trocanterea se mide desde el trocánter mayor hasta la caja y se le suma los 40 cms que mide ésta.

La tecnología aplicada a la antropometría

Dada la gran cantidad de herramientas disponibles descritas en el apartado anterior y que todas ellas son de carácter manual, parece obvio que tomar todas las medidas necesarias para realizar un estudio antropométrico puede ser muy laborioso para la persona que realiza el estudio y muy tedioso para la persona sobre la que se realiza el estudio.

Podríamos determinar tres fases en la toma de datos antropométricos utilizando la tecnología. A continuación definimos dichas fases.

- **Toma de datos.** Como ya se ha mencionado, una de las tareas más laboriosas a la hora de realizar un estudio antropométrico es la toma de datos y medidas. A parte de lo laborioso para la persona que realiza el estudio, es un proceso que, en función del objetivo del estudio, puede demorarse en gran medida resultando por ello incómodo para la persona sobre la que se realiza el estudio. El escaneado tridimensional, tecnología de la que hemos hablado en el capítulo segundo, supone una revolución en

este aspecto, permitiendo reducir la toma de datos a unos pocos segundos. En la actualidad ya se ha empezado a utilizar este tipo de técnicas, fundamentalmente utilizando un escáner laser basado en la técnica de triangulación sobre la que se ha hablado en el apartado 2.1.2 de esta memoria. Actualmente, la técnica más extendida es el uso de una línea laser como elemento de barrido, técnica que como se comentó, acelera la adquisición de datos frente al uso de un único punto laser. El contrapunto de estos aparatos es su elevado coste y, en la mayoría de los casos, su aparatosidad que dificulta su instalación. Por ello, en este trabajo se analiza la posibilidad de aplicar la tecnología TOF a esta ciencia, estudiando el caso particular de la cámara SR4000 que, con un tamaño poco mayor que una webcam y una gran velocidad en la adquisición de datos, podría suponer una reducción en costes y en tiempo para este proceso de la antropometría. Para finalizar este punto, es necesario matizar que en función del sistema de adquisición de datos que utilizemos puede ser necesaria una o varias tomas para abarcar todo el cuerpo o zona que queramos analizar.



Ilustración 71. Escaneado tridimensional de una persona.

- **Creación del modelo tridimensional.** Una vez que se ha realizado la toma de datos se pasa a crear el modelo tridimensional de la persona escaneada. Generalmente, el resultado de la toma de datos será una o varias nubes de puntos que será necesario tratar para obtener un modelo final. En función del sistema utilizado y de los procedimientos que se quieran establecer, este proceso puede ser simultaneo o casi simultaneo con el proceso de toma de datos. Esto se puede determinar

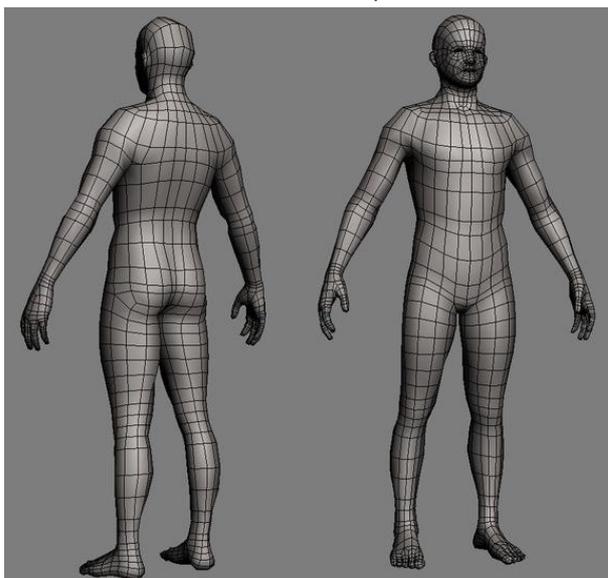


Ilustración 72. Modelo tridimensional de una persona generado por ordenador.

fundamentalmente mediante los siguientes puntos. El primero de ellos y muy a tener en cuenta es el grado de automatización que queramos aplicar en este proceso. Si bien es cierto que hay herramientas que lo realizan correctamente y de forma autónoma, algunas de las cuales se describirán más adelante, puede ser requerida la habilidad y experiencia humana a la hora de obtener un modelo tridimensional más preciso. Por otro lado, se ha comentado anteriormente que pueden ser necesarias varias tomas para adquirir todos los datos necesarios. Por esto, puede ser necesario un paso previo a la creación del modelo que se

correspondería con el proceso de combinar todas las tomas adquiridas para formar un único modelo coherente. Este paso puede ser automático utilizando algunos algoritmos como ICP, que se explicará más adelante, o utilizando un sistema de adquisición tal que permita la transformación de coordenadas para cada toma de forma que el resultado final sea el modelo completo.

- **Análisis del modelo.** Una vez que tenemos el modelo creado, podemos pasar a realizar el estudio sobre el mismo. Al ser un modelo tridimensional informático, podemos ayudarnos de todas las herramientas disponibles para realizar cualquier estudio imaginable que quisiéramos. Algunas de estas herramientas serán descritas más adelante. En cualquier caso, sobre este modelo realizaremos todas las medidas oportunas para realizar el estudio antropométrico sin necesidad de que la persona sobre la que se realiza esté presente. Se evita con esto cualquier problema derivado de un fallo de medición, en la medida en la que el sistema de adquisición de datos sea preciso, de un olvido o cualquier otro contratiempo que pudiera surgir al realizar un estudio utilizando las herramientas tradicionales.

Una vez que hemos determinado en que fases la tecnología del escaneo tridimensional puede ser útil para realizar un estudio antropométrico, hay que mencionar que no todos los datos requeridos en algunos de los estudios antropométricos pueden ser obtenidos utilizando las técnicas de escaneo tridimensional.

El primer ejemplo es el peso, dado que obviamente una cámara sea del tipo que sea no puede tomar datos acerca del peso de una persona u objeto. Es cierto que puede llegar a realizarse una estimación utilizando el volumen del modelo tridimensional y uno o varios valores medios estadísticos de densidad del cuerpo humano. Sin embargo, dado que el peso está influido por muchos factores, nunca se logrará un valor preciso. En los casos en los que ya se está aplicando la tecnología de escaneo tridimensional suele utilizarse una báscula que a su vez sirve como plataforma para que la persona se coloque a la hora de realizar el escaneo. En la ilustración de la derecha podemos observar la técnica más utilizada, el escáner laser de barrido junto con una plataforma en la que se coloca la persona. De esta forma, se pueden tomar los datos necesarios para realizar el modelo tridimensional de la persona y simultáneamente obtener el dato de su peso.

Por otra parte, hay otros datos como la impedancia bioeléctrica que por el momento deben realizarse de forma manual si bien en la mayoría de los casos sirven únicamente para

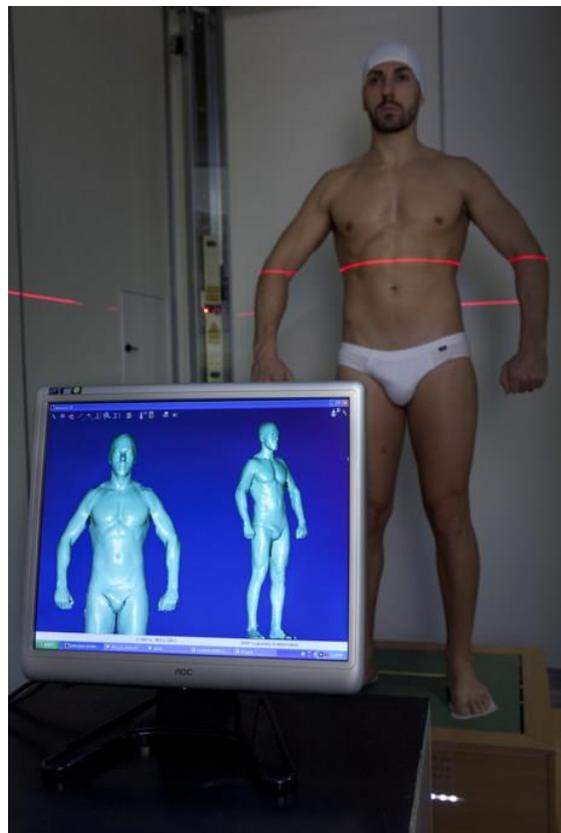


Ilustración 73. Toma de datos antropométricos mediante escaneo laser y plataforma.

un propósito muy específico por lo que quedan fuera de los estudios generales. En este caso, el análisis de impedancia bioeléctrica se basa en que el tejido magro, formado por iones en solución acuosa, conduce mejor la electricidad que el tejido graso. Por lo tanto, la resistencia corporal a la corriente eléctrica está inversamente relacionada con la masa magra. Si se conoce la masa corporal total puede estimarse la masa grasa y el porcentaje corporal graso. Visto el objetivo de este parámetro, parece obvio que su utilidad es grande en estudios relacionados con la nutrición pero no con los relacionados con industrias como el automóvil o la aeronáutica en la que este parámetro carece de importancia.

Aplicaciones de la antropometría

Ahora que conocemos los fundamentos básicos de la antropometría, podemos conocer algunas de las aplicaciones actuales en las que se aplica dicha ciencia. Dado su carácter, la antropometría puede definirse como una ciencia auxiliar por lo que por norma general sirve como base a otras ramas relacionadas que utilizan los conocimientos obtenidos mediante la antropometría con fines más prácticos destinados a usos concretos.

Ya se ha comentado anteriormente de forma superficial que uno de los campos en los que se aplica la antropometría es el de la industria. En este punto, la antropometría está intensamente relacionada con la ergonomía la cual se puede definir como la disciplina que se encarga del diseño de lugares de trabajo, herramientas y tareas, de modo que coincidan con las características fisiológicas, anatómicas, psicológicas y las capacidades del trabajador. Para ello, busca la optimización de los tres elementos del sistema humano-máquina-ambiente. Dada esta definición, parece más que evidente el hecho de que todo trabajo de ergonomía, al menos todos aquellos relacionados con los aspectos físicos, debe beber de las fuentes de la antropometría. En este campo es fundamentalmente aplicado en conceptos relacionados con los medios de transporte humano al aplicar diferentes estudios antropométricos-ergonómicos a los diseños de habitáculos y plazas destinadas a personas en los vehículos, así como el peso, factor determinante a la hora de establecer, junto con la potencia del propulsor de dicho vehículo, la capacidad del mismo. Por otra parte, el diseño de objetos de uso cotidiano o incluso la arquitectura basan parte de sus especificaciones en estudios antropométricos-ergonómicos de forma que objetos y edificios sean cómodos y saludables. Este último aspecto ha tomado gran relevancia en los últimos tiempos a la hora de determinar los espacios y elementos destinados a las zonas de trabajo e incluso hay legislación al respecto, como por ejemplo la norma ISO 9241 que determina los requisitos ergonómicos para trabajos de oficina con pantallas de visualización de datos o PVD. Esta materia suele estar reflejada de forma muy habitual en los planes de prevención de riesgos laborales.

Por otro lado, otra de las ciencias derivadas de la antropometría es la biometría, cuyo objetivo es la identificación humana utilizando, en una de sus vertientes, los rasgos físicos de la persona. Los orígenes de esta ciencia parecen remontarse hasta la China del siglo XIV pero no se puso en práctica en occidente hasta el siglo XIX. Desde sus inicios hasta una época relativamente cercana, la mayoría de procesos biométricos se realizaban de forma manual. Sin embargo, los avances tecnológicos han ido posibilitando la automatización de procesos y con ello se ha conseguido expandir el uso y capacidades de esta ciencia. Entre muchas otras opciones de la

biometría, actualmente las técnicas biométricas más relacionadas con la antropometría son el reconocimiento facial y el reconocimiento por geometría de la mano.

ANEXO IV. Proyecto de escaneado tridimensional real

Para obtener una visión inicial del método de trabajo utilizado para este tipo de proyectos, se ha tomado como referencia uno de los casos más habituales de escaneado tridimensional, el escaneado de patrimonio histórico.

En el trabajo “*Levantamiento Mediante técnicas de barrido con láser escáner 3d del puente de Vinuesa*” (5) Alejandro Martín lleva a cabo una reconstrucción tridimensional del puente



Ilustración 74. Puente romano de Vinuesa.

romano-medieval de la localidad de Vinuesa, en Soria. Su construcción puede datarse alrededor del siglo I d.C. y fue posteriormente ampliado con arcos apuntados durante la Edad Media. En la actualidad, el puente queda sumergido periódicamente a merced de las aguas del pantano de la Cuerda del Pozo, así como su calzada que unía Uxama con

Visontium, la Astúrica y la Cesaraugusta, del siglo II d.C. El puente tiene una longitud de 87 metros y una anchura de 3. La altura máxima es de 5,5m.

El procedimiento

El procedimiento descrito en el proyecto estudiado es el procedimiento general para el escaneado tridimensional en este tipo de trabajos y consiste básicamente en tres aspectos:

- Toma de datos. Para realizar la toma de datos completa es necesario establecer diferentes estacionamientos para realizar diversos escaneados con el fin de minimizar el número de huecos y zonas de sombra. Así mismo, debe existir una zona de solapamiento entre las diferentes capturas para evitar zonas sin barrido. Por otro lado, hay que tener en cuenta que el sistema de coordenadas de cada captura toma como centro el propio escáner por lo que finalmente deben transformarse todas las capturas al mismo sistema de referencia. Para esto se suelen utilizar señales de puntería de forma que sea posible enlazar, registrar y georreferenciar los distintos sistemas de coordenadas.
- Procesamiento de datos. En esta tarea se busca realizar diferentes procesos sobre los datos en bruto obtenidos de forma que adquieran coherencia y utilidad para el estudio a realizar. Entre estos procesos se encuentran el alineado de las diferentes capturas, el filtrado de datos, la segmentación de elementos, el modelado tridimensional, etc. En este caso, el procedimiento de mayor interés es el alineado que fue realizado utilizando marcas posicionales sobre la estructura de forma que son luego seleccionadas como puntos concordantes.

- Visualización de los resultados. Finalmente, una vez obtenido el modelo tridimensional podemos observar el resultado del escaneado y realizar las operaciones oportunas con dicho modelo.

El hardware y el software

Para realizar este trabajo, se utilizó el escáner laser HDS-3000 de Leica y el software propietario de la misma compañía Cyclone. En su página web (19) se puede encontrar información técnica al respecto. Entre sus características más destacadas, este dispositivo utiliza la tecnología TOF al igual que la cámara SR4000 utilizando un barrido de luz láser como radiación con la que realizar la medida. La resolución máxima de una captura es de 20.000x5.000 píxeles con un ángulo hasta 360x270 grados (posee un motor para girar sobre su base y escanear entornos). Puede capturar hasta 1.800 puntos por segundo.

Realizando una comparación con la cámara SR4000 utilizada en este proyecto, el escáner HDS-3000 es más grande, más pesado y más complicado de transportar. Hay que mencionar que el kit básico incluye una batería portátil para hacer uso del mismo en el exterior. Por el contrario, el HDS-3000 es más preciso, tiene mayor alcance y mayor resolución que la cámara SR4000. Por otra parte, la velocidad de captura de la cámara SR4000 es muy superior dado que captura 25.344 puntos por frame y es capaz de tomar hasta 25 frames por segundo. Finalmente, respecto al precio del dispositivo, es mucho más elevado en el caso del escáner HDS-3000. No se ha encontrado el precio exacto pues se adquiere por encargo directo con el fabricante pero se ha encontrado que alcanza los 12.000 euros en el mercado de segunda mano. En base a los comentarios encontrados, el precio del escáner debe rondar los 30.000 euros nuevo frente a los 3.500 euros de la cámara SR4000.



Ilustración 75. Escáner laser HDS-3000 de Leica.

Los resultados

Como podemos observar en la ilustración, los resultados obtenidos con este hardware son excepcionales, alcanzando una resolución mucho mayor y a mayor distancia que la obtenida con la cámara SR4000. El hecho de incluir información de color sobre la superficie aporta mucha más vistosidad al conjunto.

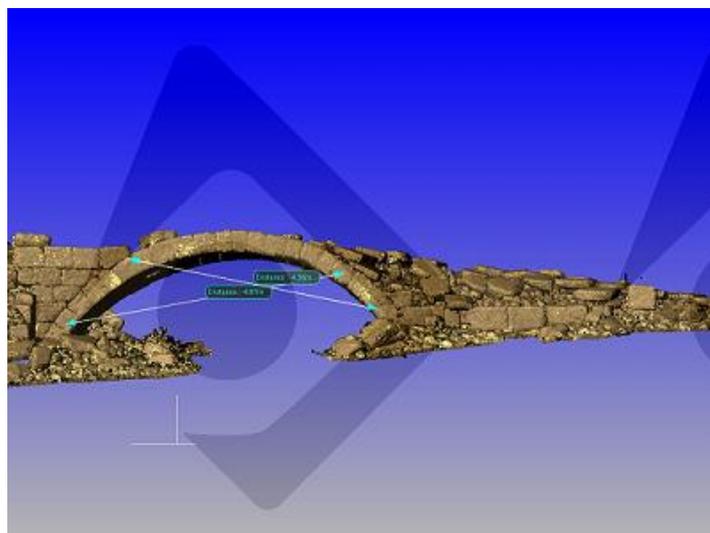


Ilustración 76. Resultados de un proyecto real.

ANEXO V. Código fuente de las pruebas

En este anexo se recoge el código fuente reseñable de las pruebas realizadas.

- **Test.cpp:** Fichero con el código de prueba necesario para ejecutar los diversos procedimientos analizados con PCL.

Test.cpp

```

#include <pcl/point_types.h>
#include <pcl/io/pcd_io.h>
#include <pcl/kdtree/kdtree_flann.h>
#include <pcl/features/normal_3d.h>
#include <pcl/surface/gp3.h>
#include <pcl/io/vtk_io.h>
#include <pcl/visualization/cloud_viewer.h>
#include <pcl/visualization/vtk.h>
#include <pcl/kdtree/kdtree_flann.h>
#include <pcl/surface/mls.h>
#include <pcl/filters/statistical_outlier_removal.h>
#include <pcl/registration/icp.h>

void test1_visualizacion(string file){
    pcl::PointCloud<pcl::PointXYZ>::Ptr cloud (new pcl::PointCloud<pcl::PointXYZ>);
    pcl::io::loadPCDFile ("files\\"+file, *cloud);
    pcl::visualization::CloudViewer viewer ("Simple Cloud Viewer");
    viewer.showCloud (cloud);
    while (!viewer.wasStopped ()){
    }
}

void test2_filtrado(string file){
    pcl::PointCloud<pcl::PointXYZ>::Ptr cloud (new pcl::PointCloud<pcl::PointXYZ> ());
    pcl::io::loadPCDFile ("files\\"+file, *cloud);
    pcl::PointCloud<pcl::PointXYZ>::Ptr cloud_filtered (new pcl::PointCloud<pcl::PointXYZ>);

    // Create the filtering object
    pcl::StatisticalOutlierRemoval<pcl::PointXYZ> sor;
    sor.setInputCloud (cloud);
    sor.setMeanK (50);
    sor.setStddevMulThresh (1.0);
    sor.filter (*cloud_filtered);

    pcl::io::savePCDFile ("files\\filtered-"+file, *cloud_filtered);
}

void test3_suavizado(string file){
    pcl::PointCloud<pcl::PointXYZ>::Ptr cloud (new pcl::PointCloud<pcl::PointXYZ> ());
    pcl::io::loadPCDFile ("files\\"+file, *cloud);

    pcl::search::KdTree<pcl::PointXYZ>::Ptr tree (new pcl::search::KdTree<pcl::PointXYZ>);
    pcl::PointCloud<pcl::PointNormal> mls_points;
    pcl::MovingLeastSquares<pcl::PointXYZ, pcl::PointNormal> mls;
    mls.setComputeNormals (true);

    // Parametros
    mls.setInputCloud (cloud);
    mls.setPolynomialFit (true);
    mls.setSearchMethod (tree);
    mls.setSearchRadius (0.1);

    mls.process (mls_points);

    pcl::io::savePCDFile ("files\\mls-"+file, mls_points);
}

void test4_triangulacion(string file){
    // Load input file into a PointCloud<T> with an appropriate type
    pcl::PointCloud<pcl::PointXYZ>::Ptr cloud (new pcl::PointCloud<pcl::PointXYZ>);
    pcl::io::loadPCDFile ("files\\"+file, *cloud);

    /* the data should be available in cloud

    // Normal estimation*
    pcl::NormalEstimation<pcl::PointXYZ, pcl::Normal> n;
    pcl::PointCloud<pcl::Normal>::Ptr normals (new pcl::PointCloud<pcl::Normal>);
    pcl::search::KdTree<pcl::PointXYZ>::Ptr tree (new pcl::search::KdTree<pcl::PointXYZ>);
    tree->setInputCloud (cloud);
    n.setInputCloud (cloud);
    n.setSearchMethod (tree);
    n.setKSearch (20);

```

```

        n.compute (*normals);
        /* normals should not contain the point normals + surface curvatures

        // Concatenate the XYZ and normal fields*
        pcl::PointCloud<pcl::PointNormal>::Ptr cloud_with_normals (new
pcl::PointCloud<pcl::PointNormal>);
        pcl::concatenateFields (*cloud, *normals, *cloud_with_normals);
        /* cloud_with_normals = cloud + normals

        // Create search tree*
        pcl::search::KdTree<pcl::PointNormal>::Ptr tree2 (new
pcl::search::KdTree<pcl::PointNormal>);
        tree2->setInputCloud (cloud_with_normals);

        // Initialize objects
        pcl::GreedyProjectionTriangulation<pcl::PointNormal> gp3;
        pcl::PolygonMesh triangles;

        // Set the maximum distance between connected points (maximum edge length)
        gp3.setSearchRadius (0.025);

        // Set typical values for the parameters
        gp3.setMu (2.5);
        gp3.setMaximumNearestNeighbors (100);
        gp3.setMaximumSurfaceAngle(M_PI/4); // 45 degrees
        gp3.setMinimumAngle(M_PI/18); // 10 degrees
        gp3.setMaximumAngle(2*M_PI/3); // 120 degrees
        gp3.setNormalConsistency(false);

        // Get result
        gp3.setInputCloud (cloud_with_normals);
        gp3.setSearchMethod (tree2);
        gp3.reconstruct (triangles);

        // Additional vertex information
        std::vector<int> parts = gp3.getPartIDs();
        std::vector<int> states = gp3.getPointStates();

        pcl::io::saveVTKFile ("files\\"+file+".vtk", triangles);

        cout << "Mesh generado.";
    }

int main (int argc, char** argv){
    int testNumber;
    do{
        cout << "Numero del test ";
        cin >> testNumber;
        if(testNumber==1){
            cout << "Fichero a visualizar: ";
            string file;
            cin >> file;
            test1_visualizacion(file);
        }else if(testNumber==2){
            cout << "Fichero a filtrar: ";
            string file;
            cin >> file;
            test2_filtrado(file);
        }else if(testNumber==3){
            cout << "Fichero a suavizar: ";
            string file;
            cin >> file;
            test3_suavizado(file);
        }else if(testNumber==4){
            cout << "Fichero a triangular: ";
            string file;
            cin >> file;
            test4_triangulacion(file);
        }
    } while (testNumber != 0);
    return (0);
}

```

Bibliografía

- [1] D. Sánchez del Álamo Benguigui, 'Time Of Flight Camera.' Universidad Carlos III de Madrid, 2010.
- [2] P. Cignoni, 'MeshLab.' [Online]. Available: <http://meshlab.sourceforge.net/>. [Accessed: 25-Oct-2015].
- [3] 'PCL - Point Cloud Library (PCL)', 26-Sep-2013. [Online]. Available: <http://pointclouds.org/>. [Accessed: 25-Oct-2015].
- [4] 'SwissRanger | Heptagon.' [Online]. Available: <http://www.mesa-imaging.ch/products/sr4000/>. [Accessed: 25-Oct-2015].
- [5] A. Martín Romo, 'Levantamiento Mediante técnicas de barrido con láser escáner 3d del puente de Vinuesa.' Universidad Politécnica de Madrid, 2008.
- [6] 'XNA Game Studio 4.0', 2012. [Online]. Available: [http://msdn.microsoft.com/es-es/library/bb200104\(v=xnagamestudio.40\).aspx](http://msdn.microsoft.com/es-es/library/bb200104(v=xnagamestudio.40).aspx). [Accessed: 25-Oct-2015].
- [7] J. M. Mastre Rodríguez, 'RECONSTRUCCIÓN TOPOGRÁFICA HUMANA MEDIANTE TIEMPO DE VUELO (ToF).' UPM-INEF, 2011.
- [8] G. Sirat and D. Psaltis, 'Conoscopic Holography', pp. 324–330, Jun. 1985.
- [9] M. Corsini, P. Cignoni, and R. Scopigno, 'Efficient and Flexible Sampling with Blue Noise Properties of Triangular Meshes', *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 6, pp. 914–924, Jun. 2012.
- [10] P. J. Besl and H. D. McKay, 'A method for registration of 3-D shapes', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [11] 'Kinect para Xbox One.' [Online]. Available: <http://www.xbox.com/es-ES/xbox-one/accessories/kinect-for-xbox-one>. [Accessed: 25-Oct-2015].
- [12] 'Morfología matemática', *Wikipedia*. Wikipedia, 2005.
- [13] P. Cignoni, 'MeshLab: an Open-Source Mesh Processing Tool', in *Eurographics Italian Chapter Conference*, 2008.
- [14] 'gnu.org', 08-Nov-2014. [Online]. Available: <http://www.gnu.org/copyleft/gpl.html>. [Accessed: 25-Oct-2015].
- [15] 'The BSD 2-Clause License | Open Source Initiative.' [Online]. Available: <http://opensource.org/licenses/bsd-license.php>. [Accessed: 25-Oct-2015].
- [16] S. -Y, G. Dang, K. Xu, B. Li, Wang, Y. -Z, Z. -Q, Z.-Q. Cheng, and Y.-Z. Wang, 'A Survey of Methods for Moving Least Squares Surfaces', 2008.
- [17] '3-D scanning methodologies for internationally compatible anthropometric databases', no. ISO 20685:2010. ISO, 2013.
- [18] 'Microsoft Support Lifecycle.' [Online]. Available: <http://support.microsoft.com/lifecycle/search/>. [Accessed: 25-Oct-2015].
- [19] 'Leica HDS3000 - Leica Geosystems', 07-Dec-2006. [Online]. Available: <http://hds.leica-geosystems.com/en/5574.htm>. [Accessed: 25-Oct-2015].