

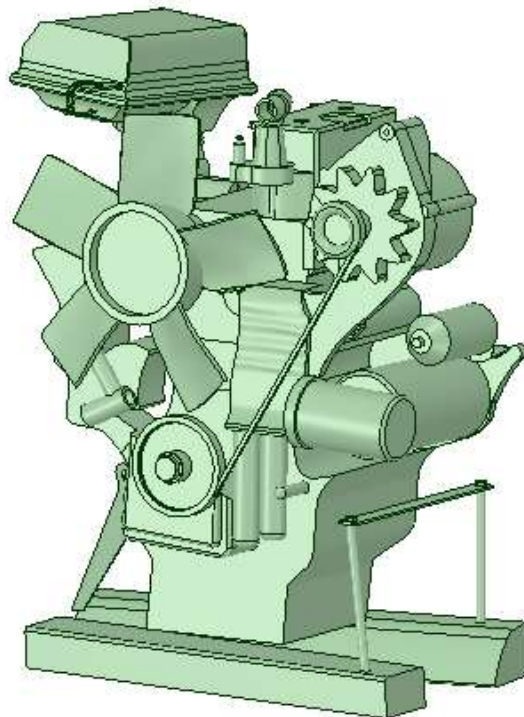


VNIVERSIDAD
D SALAMANCA

ESCUELA POLITÉCNICA SUPERIOR DE ÁVILA
MÁSTER GEOTECNOLOGÍAS CARTOGRÁFICAS EN INGENIERÍA Y
ARQUITECTURA

PROYECTO FINAL DE MÁSTER

Procesos de ingeniería inversa en elementos de
automoción usando Gaming Sensors



AUTOR: Beatriz Paniagua Rivero

TUTORES: Pablo Rodríguez González / Diego González Aguilera

FECHA: Septiembre 2014

ÍNDICE:

MEMORIA

| | |
|--|-----------|
| 1. INTRODUCCIÓN..... | 1 |
| 1.1 Contexto del trabajo desarrollado..... | 3 |
| 1.2 Objetivos perseguidos..... | 3 |
| 2. INSTRUMENTACIÓN. SOFTWARE. MEDIOS EMPLEADOS..... | 5 |
| 2.1 Instrumentación..... | 5 |
| 2.1.1 Kinect 2..... | 5 |
| 2.1.2 Brazo robótico..... | 8 |
| 2.2 Software..... | 10 |
| 2.3 Medios empleados..... | 11 |
| 3. METODOLOGÍA SEGUIDA..... | 13 |
| 3.1 Trabajo de campo..... | 13 |
| 3.1.1 Planificación..... | 13 |
| 3.1.2 Toma de datos Kinect 2..... | 14 |
| 3.1.3 Toma de datos brazo robótico..... | 17 |
| 3.2 Trabajo de oficina..... | 20 |
| 3.2.1 Procesamiento datos Kinect 2..... | 20 |
| 3.2.2 Procesamiento datos brazo robótico..... | 31 |
| 4. COMPARACIÓN EQUIPOS..... | 36 |
| 5. RESULTADOS..... | 45 |
| 5.1 Nubes de puntos..... | 45 |
| 5.2 Mallas poligonales..... | 48 |
| 5.3 Comparaciones..... | 53 |

| | |
|--|-----------|
| 5.3.1 Nube de puntos individual (Kinect) vs. Nube de puntos global (Kinect) (NI(K)-NG(K))..... | 53 |
| 5.3.2 Nube de puntos (Kinect) vs. Malla modelado 3D..... | 60 |
| 5.3.3 Malla brazo robótico vs. Malla modelado 3D..... | 62 |
| 5.3.4 Nube de puntos (Kinect) vs. Malla brazo robótico..... | 65 |
| 6. CONCLUSIÓN..... | 68 |
| 7. BIBLIOGRAFÍA..... | 70 |
| 8. AGRADECIMIENTOS..... | 73 |

ANEXOS

| | |
|----------------------|-----------|
| ANEXO I..... | 75 |
| ANEXO II..... | 77 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1. Kinect 2..... | 5 |
| Figura 2. Funcionamiento escáner láser..... | 8 |
| Figura 3. Romer Absolute Arm con escáner integrado..... | 9 |
| Figura 4. Toma de datos Kinect 2..... | 14 |
| Figura 5. Archivo ASCII..... | 15 |
| Figura 6. Imagen capturada por el sensor Kinect 2..... | 16 |
| Figura 7. Escaneado brazo robótico..... | 17 |
| Figura 8. Captura y configuración..... | 18 |
| Figura 9. Entrada datos CloudCompare..... | 21 |
| Figura 10. Nube de puntos con ruido..... | 22 |
| Figura 11. Alineamiento nubes de puntos 1ª fase..... | 24 |
| Figura 12. Alineamiento nube de puntos 2ª fase..... | 25 |
| Figura 13. Malla 10% creada con RealWorks..... | 26 |
| Figura 14. Reducir ruido..... | 27 |
| Figura 15. Envolver..... | 27 |
| Figura 16. Rellenar..... | 27 |
| Figura 17. Malla creada por Geomagic..... | 27 |
| Figura 18. Dividir cara..... | 29 |
| Figura 19. Creación ventilador..... | 29 |
| Figura 20. Reconstrucción 3D..... | 31 |
| Figura 21. Color según pasada..... | 32 |
| Figura 22. Best fit align..... | 34 |
| Figura 23. Generación modelo poligonal IMMerge..... | 35 |
| Figura 24. Alineación nubes..... | 37 |

| | |
|---|----|
| Figura 25. Ventana de resultados aproximados..... | 39 |
| Figura 26. Tipos de modelización..... | 39 |
| Figura 27. Búsqueda vecino más cercano..... | 40 |
| Figura 28. Diagrama Voronoi..... | 42 |
| Figura 29. Triangulación Delaunay..... | 42 |
| Figura 30. Comando Matlab..... | 44 |
| Figura 31. Vista frontal nube puntos Kinect..... | 45 |
| Figura 32. Vista trasera nube puntos Kinect..... | 45 |
| Figura 33. Perfil izquierdo nube puntos Kinect..... | 46 |
| Figura 34. Perfil derecho nube puntos Kinect..... | 46 |
| Figura 35. Planta nube puntos Kinect..... | 46 |
| Figura 36. Nube puntos brazo robótico..... | 47 |
| Figura 37. Detalle nube de puntos brazo robótico..... | 48 |
| Figura 38. Vista frontal malla..... | 49 |
| Figura 39. Vista trasera malla..... | 49 |
| Figura 40. Vista lateral frontal malla..... | 49 |
| Figura 41. Vista lateral trasera malla..... | 49 |
| Figura 42. Vista frontal sólido 3D..... | 50 |
| Figura 43. Vista trasera sólido 3D..... | 50 |
| Figura 44. Vista lateral frontal sólido 3D..... | 50 |
| Figura 45. Vista lateral trasera sólido 3D..... | 50 |
| Figuras 46 y 47. Malla base y sólido 3D..... | 51 |
| Figura 48. Modelo poligonal motor..... | 52 |
| Figura 49. Vista frontal modelo brazo robótico..... | 52 |
| Figura 50. Vista en planta modelo brazo robótico..... | 52 |
| Figura 51. Información gráfica error (None)..... | 55 |

| | |
|---|----|
| Figura 52. Histograma de error (None)..... | 55 |
| Figura 53. Valores de error gaussianos (None)..... | 55 |
| Figura 54. Información gráfica error (Least Square Plane)..... | 56 |
| Figura 55. Histograma de error (Least Square Plane)..... | 56 |
| Figura 56. Valores de error gaussianos (Least Square Plane)..... | 56 |
| Figura 57. Información gráfica error (2D ½ Triangulation)..... | 57 |
| Figura 58. Histograma error (2D ½ Triangulation)..... | 57 |
| Figura 59. Valores de error gaussianos (2D ½ Triangulation)..... | 57 |
| Figura 60. Información gráfica error (Height Function)..... | 58 |
| Figura 61. Histograma error (Height Function)..... | 58 |
| Figura 62. Valores de error gaussianos (Height Function)..... | 58 |
| Figura 63. Información gráfica del error Nube vs. Modelado..... | 60 |
| Figura 64. Histograma error (Nube-Modelado)..... | 61 |
| Figura 65. Valores de error gaussianos (Nube-Modelado)..... | 61 |
| Figura 66. Alineación mallas..... | 62 |
| Figura 67. Información gráfica del error Malla brazo vs. Modelado..... | 63 |
| Figura 68. Histograma error (Malla brazo-Modelado)..... | 64 |
| Figura 69. Valores de error gaussianos (Malla brazo-Modelado)..... | 64 |
| Figura 70. Alineación nube y malla..... | 65 |
| Figura 71. Información gráfica del error Nube Kinect vs. Malla brazo..... | 66 |
| Figura 72. Histograma error (Nube Kinect-Malla brazo)..... | 67 |
| Figura 73. Valores de error gaussianos (Nube Kinect-Malla brazo)..... | 67 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1. Comparativa Kinect vs. Kinect 2..... | 6 |
| Tabla 2. Resultados aproximados (NI(K)-NG(K))..... | 54 |
| Tabla 3. Estimaciones robustas..... | 54 |
| Tabla 4. Valores gaussianos NONE..... | 55 |
| Tabla 5. Valores gaussianos LEAST SQUARE PLANE..... | 56 |
| Tabla 6. Valores gaussianos 2D ½ TRIANGULATION..... | 57 |
| Tabla 7. Valores gaussianos HEIGHT FUNCTION..... | 58 |
| Tabla 8. Resultados aproximados (N(K)-Modelo)..... | 60 |
| Tabla 9. Estimaciones robustas (Nube-Modelado)..... | 60 |
| Tabla 10. Resultados aproximados (Malla brazo-Modelo)..... | 63 |
| Tabla 11. Resultados aproximados (Nube-Malla brazo)..... | 66 |
| Tabla 12. Estimaciones robustas (Nube-Malla brazo)..... | 66 |

ÍNDICE DE GRÁFICOS

Gráfico 1. Error de precisión Kinect.....7

MEMORIA

1. INTRODUCCIÓN

Desde hace varias décadas, en el mundo de la ingeniería se ha venido incrementando el interés por el desarrollo de una metodología que permita relacionar el mundo real con el teórico. Esta estrategia se lleva a cabo mediante la conocida ingeniería inversa, la cual se vale de objetos de referencia reales para obtener modelos o duplicados a partir de los mismos. Se trata de un *proceso de diseño que puede ser aplicado a un producto existente, a un prototipo o un concepto detallado. Utiliza una variedad de técnicas como modelos, esquemas o teorías normativas, que proveen la comprensión total del producto* [1].

De acuerdo con [2], se define como *aquel proceso analítico-sintético que busca determinar las características y/o funciones de un sistema, máquina o producto o parte del mismo. El propósito, por tanto, es determinar un modelo de un objeto o producto o sistema de referencia. Del mismo modo, se puede interpretar como [3] el proceso de diseñar un sustituto, el cual reemplace de forma aceptable a un producto. Se fundamenta en el análisis de un ejemplar y se aplica cuando el proceso de diseño o la documentación no está disponible.*

A diferencia de la ingeniería tradicional, en la que se obtienen las piezas a partir de unos planos y documentos en los que se especifican detalladamente sus características, la ingeniería inversa, como su propio nombre indica, realiza el proceso contrario. Partiendo de una pieza real ya creada, se centra en la determinación del modelo del objeto de referencia. Esta tecnología resulta muy útil tanto en el duplicado de piezas, como en la modificación o rediseño de las mismas.

La ingeniería inversa presenta un amplio abanico de posibilidades en diferentes campos, como son el del mantenimiento de maquinaria, sustitución de componentes o las innovaciones tecnológicas. Permiten mejorar la calidad del producto final, ya que se pueden hacer comprobaciones de precisión. Así mismo, posibilita la fabricación de determinados componentes que estén dañados, sin la necesidad de comprar el equipo entero. Otra de las ventajas que ofrece es el ahorro de tiempo y dinero en la creación de nuevos diseños y el lanzamiento de nuevos productos al mercado. Es una valiosa solución para extraer las dimensiones de modelos o prototipos que siguen formas libres.

Aunque este proyecto se centra en la aplicación de la ingeniería inversa desde un punto de vista industrial y de investigación y desarrollo, existen otros muchos sectores en los que representa una gran utilidad, como es el caso del campo paleontológico, automovilístico, médico, aeroespacial, energético o de control de calidad.

El objetivo final de la ingeniería inversa es la recreación del diseño de un modelo de referencia. Para ello, será necesario contar con los medios adecuados de captura que permitan obtener toda la información necesaria para crear un modelo CAD de la pieza. Dentro del mercado, se encuentran diferentes soluciones de captura, en función del precio, calidad y precisión del medio empleado. Es en este punto en el que hay que decidir que opción se adecúa mejor a nuestras pretensiones y objetivos, confrontando el coste y los resultados finales deseados.

En este contexto, en el presente proyecto se realizará la comparación entre dos medios de captura totalmente opuestos, como es el uso de un Gaming Sensor y de un brazo robótico especializado en este ámbito.

Los dispositivos de bajo coste representan una alternativa de interés general en el modelado 3D de múltiples aplicaciones de ingeniería y arquitectura en las que el grado de precisión necesaria no sea excesivo. Estos sistemas permiten llevar a cabo procesos de escaneo y modelado 3D con un coste muy reducido y unos resultados muy aceptables.

REFERENCIAS

[1] Lefever, D. L. Wood K. L. *Design for assembly techniques in reverse engineering and redesign*. Proceedings of the ASME Design Engineering technical conferences and design theory methodology conference, 18-22 Agosto 1996. Irvine, California

[2] Jiménez E., Reyes L., García A. *Algunas consideraciones sobre la Ingeniería Inversa, Informe Interno de Investigación*. Centro de Tecnología Avanzada de ITESCA, Red Alfa, 2006, Sonora, México.

[3] Borja V. *Redesign Supported by Data Models with Particular Reference to Reverse Engineering*. PhD Thesis, 1997, Loughborough University.

1.1 Contexto del trabajo desarrollado

La ingeniería inversa está considerada como una de las técnicas que se encuentran en mayor auge en los últimos tiempos. Cabe destacar el enorme margen de mejora y desarrollo que aún existe. Siguiendo estas líneas, surge la idea de realizar un análisis exhaustivo entre diferentes medios de captura, con su posterior procesado y resultados finales obtenidos.

Estas evaluaciones nos permitirán plantearnos qué tipo de mejoras y avances se pueden ir implementando en el desarrollo de la ingeniería inversa. Dado el elevado coste económico que supone poseer un equipo de captura de datos altamente cualificado, se hace necesario investigar y profundizar en nuevas técnicas e instrumentos que ofrezcan resultados óptimos con un coste notablemente inferior.

Es por ello, por lo que en este proyecto se trata de comparar dos tecnologías completamente distintas. Por un lado, se cuenta con una solución de bajo coste, como es un Gaming Sensor, concretamente la Kinect 2 de Microsoft. Tanto la captura de datos, como el procesamiento de los mismos y su resultado final, serán contrastados con los desarrollados por un brazo robótico. En este caso se trata de un Romer Absolute Arm de la casa Hexagon.

Para llevar a cabo el análisis y poder plasmar las diferencias entre los dos medios, se ha utilizado un motor de un automóvil de la marca Kubota. Las diferentes curvaturas y elementos que posee, permitirán contemplar y discernir las similitudes y disparidades registradas por ambos instrumentos. El motor ha sido instalado en las dependencias del grupo TIDOP en la Escuela Politécnica Superior de Ávila. De este modo, quedarán establecidas unas condiciones de iluminación interior en el momento de la toma de datos, cuestión importante a tener en cuenta tanto durante la misma toma de datos, como en el caso de que fuesen necesarias otras futuras.

Tanto la instrumentación de captura como el software necesario para el procesamiento han sido proporcionados por la Escuela Politécnica Superior de Ávila (Universidad de Salamanca).

1.2 Objetivos perseguidos

El principal objetivo es realizar una comparación entre dos instrumentos de captura distintos. Uno de ellos es de bajo coste y con una finalidad inicialmente distinta a la que se está aplicando en este proyecto. Mientras que el otro requiere una inversión mayor, pero está completamente destinado a la realización de esta labor.

Tras el análisis, que comienza desde el momento de la captura, habrá que determinar y concluir hasta qué punto resulta rentable o adecuado un medio u otro. Para ello, es imprescindible tener en cuenta la finalidad y el uso que queremos darle.

Para poder alcanzar dicha conclusión se fija como objetivo la obtención de una nube de puntos de la misma región del motor, por ambos equipos, pudiendo así realizar una comparación entre las mismas para determinar los errores generados, tanto por el propio aparato, como la propagación de errores en el alineamiento. Además, esta equiparación también tendrá lugar a nivel de malla.

Por otro lado, se reconstruirá el modelo del motor en 3D a partir de los datos obtenidos por el Gaming Sensor. Dada la enorme cantidad de detalles que presenta el objeto analizado, resulta imposible la completa recreación del mismo, por lo que únicamente se llevará a cabo en aquellas zonas y elementos más visibles. De este modo, también se analiza hasta qué nivel de detalle es capaz de llegar el instrumento de bajo coste consiguiendo unos resultados óptimos.

Por tanto, se llevarán a cabo las siguientes comparaciones:

- Nube de puntos individual Kinect vs. Nube de puntos global Kinect.
- Nube de puntos Kinect vs. Malla modelado 3D.
- Malla brazo robótico vs. Malla modelado 3D.
- Nube de puntos Kinect vs. Malla brazo robótico.

A su vez, las confrontaciones entre las diferentes nubes de puntos se evaluarán mediante distintos métodos, como son búsqueda del vecino más cercano, ajuste por mínimos cuadrados, triangulación de Delaunay y función de altura cuadrática.

Por tanto, a partir de los análisis desarrollados, se establece como objetivo determinar las diferencias significativas que pudieran existir entre una nube individual y el conjunto final alineado. Estos valores, que se obtendrán más adelante, serán comparados con el valor esperado, que es de unos $\pm 4,6$ mm.

2. INSTRUMENTACIÓN. SOFTWARE. MEDIOS EMPLEADOS

2.1 Instrumentación

2.1.1 Kinect 2

En la toma de datos se han empleado dos tecnologías totalmente diferentes. Por un lado, se cuenta con la solución de bajo coste, como es la KINECT 2. Dicha versión acaba de salir recientemente al mercado. En concreto, la que se ha utilizado en este proyecto corresponde a una versión de prueba adquirida por los desarrolladores.

La Kinect 2 es un sistema desarrollado por Microsoft para la videoconsola XBOX One que permite a los usuarios controlar e interactuar con la consola sin necesidad de utilizar mandos. A través de una serie de cámaras reconstruye modelos 3D en tiempo real. Un proyector emite un patrón de puntos conocidos, y en función de cómo difieren dichos puntos respecto al patrón se reconstruye la escena.



Figura 1. Kinect 2

La primera Kinect que salió al mercado constaba de tres puntos básicos. Por un lado, una cámara RGB encargada de obtener los colores de la escena y por otro, dos sensores cuya misión es conseguir la geometría y profundidad de los objetos, trabajando una de ellos por el canal infrarrojo. El cálculo de la profundidad está basado en el paralaje. Cuanto más cerca esté el punto, mayor será su paralaje y viceversa, cuanto más alejado se encuentre, será menor.

Uno de los inconvenientes de este primer modelo es que únicamente puede trabajar en interiores, puesto que con una mayor iluminación se satura. La última versión de este sistema, la Kinect 2, sustituye la cámara de IR por un emisor láser. La reconstrucción en 3D está basada en el tiempo de vuelo. Un rayo láser es emitido en fase con respecto a una de las dos mitades de píxel por el proyector, y cada píxel del detector se encarga de absorber los fotones de dicho rayo. Cada mitad de píxel se enciende y apaga rápidamente en un ciclo de 10 GHz. A medida que el tiempo de recorrido del láser aumenta, se va concentrando una mayor cantidad de fotones que llega cuando la segunda mitad de píxel está encendida. Para determinar la distancia, se comparará la cantidad de fotones recogidos en una mitad y en otra.

Los fotones reflejados en objetos situados a mayor profundidad, tienen posibilidad de ser recogidos en la primera mitad de píxel. Esto generará ambigüedades, que se traducen en una pérdida de detalle, como podemos observar visualmente. Para solventar este problema podemos aumentar el tiempo en el que permanecen abiertos los píxeles.

Las principales fuentes de error residen en la baja reflectividad del material (aquellos de tonos oscuros o negro), excesiva iluminación, que saturará la información, superficies inclinadas donde se generarán distorsiones y las discontinuidades espaciales, que provocarán problemas de matching entre los puntos. Todos estos aspectos han de ser tenidos en cuenta, ya que nos los vamos a encontrar durante el desarrollo del proyecto.

La Kinect toma dos tipos de medidas, una de baja resolución sin ambigüedades en la distancia, y otra de alta basada en la anterior para poder eliminar estos problemas.

El sensor incorpora un sistema de rechazo de luz ambiental, por lo que no se ve afectado por la iluminación externa. Detecta para cada píxel individual, cuándo está saturado, y lo resetea a la mitad de exposición. Esto nos abrirá un amplio abanico de posibilidades, ya que permitirá realizar capturas de datos en ambientes exteriores.

A continuación se incluye una comparación de las especificaciones entre los dos modelos:

| | Xbox 360 Kinect Sensor | Kinect for Windows v2 |
|-----------------------------|--|---|
| Field of View (FOV) | 57.5º horizontal by 43.5º vertical | 70º horizontal by 60º vertical |
| Resolvable Depth | 0.8 m -> 4.0 m | 0.5 m -> 4.5 m |
| Color Stream | 640 x 480 x 24 bpp 4:3 RGB @ 30 fps 640 x 480 x 16 bpp 4:3 YUV @ 15 fps | 1920 x 1080 x 16 bpp 16:9 YUY2 @ 30 fps |
| Depth Stream | 320 x 240 x 16 bpp, 13-bit depth | 512 x 424 x 16 bpp, 13-bit depth |
| Infrared (IR) Stream | No IR stream | 512 x 424, 11-bit dynamic range |
| Registration | Color <-> depth | Color <-> depth and active IR |
| Audio Capture | 4-mic array returning 48 Hz audio | 4-mic array returning 48 KHz audio |
| Data Path | USB 2.0 | USB 3.0 |
| Latency | ~90 ms with processing | ~60 ms with processing |
| Tilt Motor | Vertical only | No tilt motor |

Tabla 1. Comparativa Kinect vs. Kinect 2

Para poder trabajar correctamente con la Kinect 2, el equipo que se utilice ha de poseer unos requerimientos mínimos, la mayoría de ellos, al alcance de cualquier de cualquier ordenador portátil actual. Los requerimientos del hardware son:

- Procesador de 64-bit (x64).
- Procesador dual-core 3.1GHz.
- Puerto USB 3.0 para la conexión de la Kinect 2.
- 2 GB de RAM.
- Tarjeta gráfica que soporte DirectX 11 (Intel HD 4000, AMD Radeon HD 6470M, NVIDIA Geforce 610m, AMD Radeon HD 6570).
- Windows 8 ó 8.1.

En cuanto a los errores generados durante la captura de objetos no existen datos precisos relativos a la Kinect 2, ya que ha salido al mercado hace tan sólo un par de meses. Por ello, se le puede suponer el mismo error que a la Kinect 1 ó incluso ligeramente inferior.

En el siguiente gráfico se muestra la relación entre la distancia al objeto y el error producido.

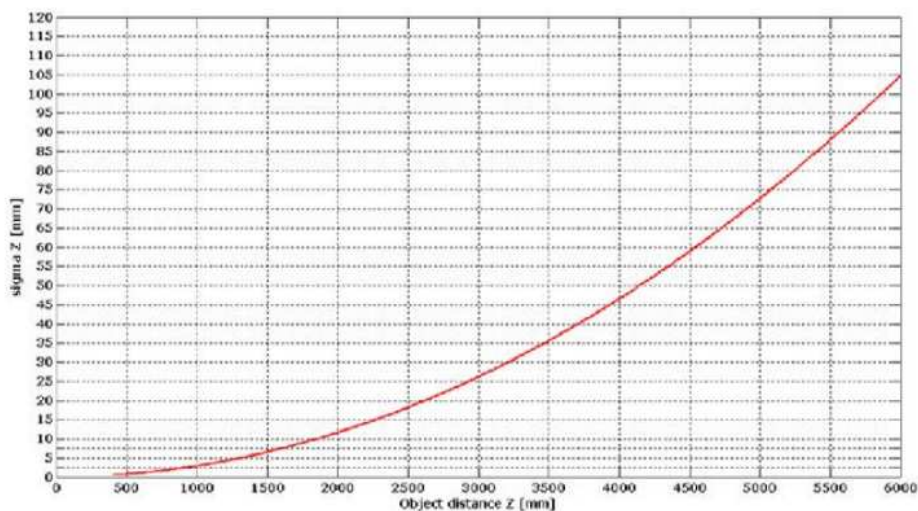


Gráfico 1. Error de precisión de Kinect [4]

A la hora de predecir el error que se generará se tendrá en cuenta la distancia desde la cual se realicen las tomas.

REFERENCIAS

[4] Khoshelham, K.; Elberink, S.O. *Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications*. Sensors 2012, 12 1437-1454.

2.1.2 Brazo robótico

En contraposición con el gaming sensor de bajo coste, se empleará un brazo robótico especializado para llevar a cabo una toma de datos sobre el mismo motor. En este caso, el equipo utilizado ha sido un ROMER ABSOLUTE ARM 7325 SI de la casa comercial Hexagon.

Se trata de una herramienta de metrología con múltiples aplicaciones en distintos sectores de la industria (digitalización 3D, modelado 3D, inspección de nubes de puntos, ingeniería inversa, etc.). Cuenta con una amplia gama de sensores de contacto y sin contacto y escáneres láser, lo que permite al usuario escoger la mejor solución en función de sus necesidades. En este caso, dado que el objeto de estudio presenta una forma irregular y complicada, se opta por el uso del escáner láser integrado en el equipo.

El funcionamiento del sistema de captura está basado en el barrido del plano de triangulación mediante un espejo oscilante. El escáner emite una luz láser para examinar el objeto y ésta es capturada posteriormente por una cámara, con lo que queda determinada la posición. El emisor láser, el punto del objeto donde se refleja y la cámara que lo recibe, forman un triángulo, donde la distancia entre emisor y cámara y el ángulo que forma el rayo, son conocidos. De esta manera, se obtiene la forma y tamaño del triángulo generado, determinando así la posición tridimensional de cada punto del objeto.

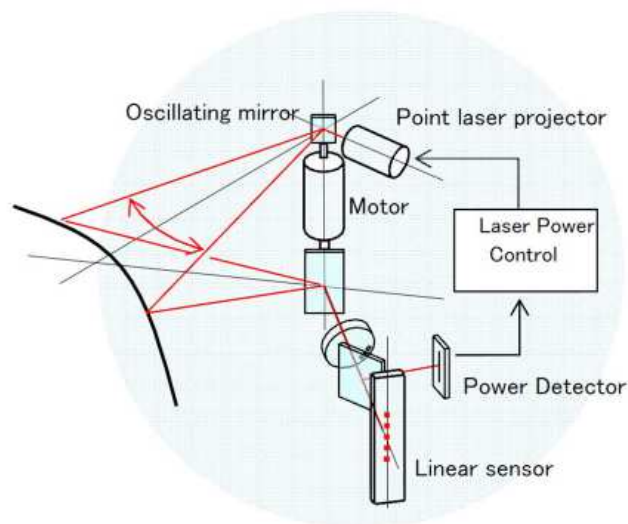


Figura 2. Funcionamiento escáner láser

Al realizar un barrido sobre el plano y no sólo capturar un único punto, se mejora la velocidad de toma de datos. Esta velocidad de barrido afectará a la densidad de puntos capturada. En función de la velocidad de movimiento del usuario, se

determinará la distancia entre las líneas de escaneo. En cada instante, sólo una fracción de la escena es visible, por lo que el sistema no se verá perjudicado ante las condiciones de iluminación externa. La intensidad del láser se adapta en tiempo real de forma continua durante las mediciones. El control de potencia láser semiautomático hace que sea posible trabajar sobre diferentes tipos de superficie. Además, este sistema de escaneado reduce el ruido causado por el speckle (patrón de interferencia que se produce cuando un haz de luz coherente, como de un láser, se refleja difusamente sobre una superficie con una estructura rugosa).

Finalmente, la nube de puntos capturada se convierte en una malla poligonal, representando así el modelo en 3D. La elevada densidad de la nube de puntos permite generar un mapa de desviaciones.

A parte del propio brazo robótico, con el correspondiente escáner integrado, otros componentes adicionales que forman parte del equipo son:

- Sondas (de diferentes tamaños).
- Piezas de calibración.
- Trípode y base de sujeción (imprescindibles para poder establecer un sistema de coordenadas).
- Base magnética.



Figura 3. Romer Absolute Arm con escáner integrado

Una de las ventajas que ofrece este brazo robótico es que incluye la opción del sistema wireless, con el cual se puede trabajar sin la necesidad de cables. Esto permite a su vez una rotación infinita, pudiendo así capturar todo tipo de formas libres. La posibilidad de conectarlo a la corriente mientras se cargan las baterías, ahorrará tiempo en las sesiones de captura prolongadas. La duración estimada de autonomía es de 8 horas, para el palpador, y 2,5 horas de escaneado continuo.

Las distintas características y especificaciones tanto del brazo robótico como del escáner se incluyen en el anexo I.

Tal y como se muestra en el anexo I, la precisión del sistema de escaneo es de $\pm 0,084$ mm.

2.2 Software

Dentro del software se debe distinguir aquel necesario para la captura de datos, específico para cada uno de los equipos, y el del posterior procesamiento.

En primer lugar, para la captura de datos de la Kinect 2, son necesarios los drivers (k4w o Kinect for Windows) y un software que haga uso de esos drivers para la captura. En cuanto a las prestaciones del equipo, es necesario poseer el sistema operativo Windows 8 con una conexión USB 3.0. Estos requerimientos son imprescindibles.

La eliminación del ruido de las distintas capturas se realiza con CLOUDCOMPARE V2, programa que posteriormente se utilizará también para el análisis de errores. CloudCompare es un software de procesamiento de nubes de puntos en 3D. Aunque originalmente está diseñado para la comparación entre dos nubes de puntos, o entre una nube de puntos y una malla, también ofrece la posibilidad de editar dichas nubes.

Una vez que se ha eliminado todo el ruido posible, se procede al alineamiento de las tomas y la generación de la malla con REALWORKS SURVEY ADVANCED 6.4. Este software desarrollado por Trimble permite trabajar con datos tomados desde otras marcas distintas a la de la propia casa comercial. Permite visualizar, explorar, registrar y manipular diferentes nubes de puntos, así como exportarlas a AutoCAD o MicroStation, entre otros.

A partir de esta malla se reconstruirá el modelo en 3D del motor mediante GEOMAGIC DESIGN DIRECT. Combina el procesamiento de datos escaneados, diseño CAD, edición de mallas y puntos 3D y modelado. Esta herramienta permite extraer las distintas superficies a partir de una malla, o también generar la malla partiendo de una

nube de puntos. Tras la generación del modelo tridimensional, éste será exportado en formato STL.

En cuanto al brazo robótico Romer, se puede utilizar el software proporcionado por el fabricante (Romer RDS) para la captura. Sin embargo, en esta ocasión se ha empleado POLYWORKS ALIGN V12.1, ya que todos los software de procesamiento tienen un módulo específico de captura. Polyworks incorpora el proceso completo, desde la captura de las nubes de puntos hasta el procesamiento y modelado de superficies. Por ello, una vez que se encuentran alineadas y optimizadas las capturas, se genera el modelo poligonal en formato STL con el módulo IMMerge.

Tras el procesamiento de todos los datos capturados, se procede al análisis de los mismos. Las distintas comparaciones entre nubes de puntos, nubes frente a mallas y las poligonales obtenidas por ambos equipos, serán efectuadas con CloudCompare.

En un último lugar, y como complemento en el proceso de comparación entre nubes, se utiliza MATLAB. A través de esta herramienta de programación se calculan los valores de la mediana y la diferencia absoluta de medianas, que se corresponden con las estimaciones robustas de la media y desviación típica respectivamente. Estos datos serán los que se evaluarán a la hora de analizar los errores generados.

Matlab es un software diseñado para el cálculo numérico, la visualización y la programación. Permite analizar datos, desarrollar algoritmos y crear modelos o aplicaciones.

2.3 Medios empleados

Durante el desarrollo de este proyecto han sido necesarios diversos medios. A parte de los ya comentados instrumentos de captura, Kinect 2 y Romer Absolute Arm 7325 SI, son indispensables otros equipos, tanto para la toma de datos como para el procesamiento de los mismos.

Para la captura de datos de la Kinect se ha utilizado una TABLET LENOVO YOGA 2, con procesador i5, gráfica integrada y 4 Gb de RAM. Los únicos requisitos que debe cumplir son el sistema operativo Windows 8 y la conexión USB 3.0.

En cambio, para la captura y procesamiento de los datos del brazo robótico, no es estrictamente necesario poseer determinadas prestaciones a nivel de sistema operativo o medios de entrada. Este apartado se ha resuelto con una WORKSTATION Z400 con un procesador Intel Xeon w3550, 6 Gb de RAM y tarjeta gráfica Nvidia Quadro 4000 de 2 Gb.

El resto del procesamiento se ha llevado a cabo con un TOSHIBA SATELLITE L750/L755, con procesador Intel Core i7, tarjeta gráfica Nvidia Geforce y 4 Gb de RAM. Aunque se trata de un ordenador con unas prestaciones más que suficientes para la correcta ejecución de los distintos programas, en algunas ocasiones ha sido necesario limitar el número de puntos con los que se trabajaba, dado el enorme esfuerzo y tiempo de desarrollo que precisaba.

3. METODOLOGÍA SEGUIDA

3.1 Trabajo de campo

3.1.1 Planificación

Antes de comenzar la captura de los datos es indispensable planificar correctamente todos los pasos que se van a dar, ya que esta pequeña labor previa, ayudará a reducir el tiempo de desarrollo del trabajo y a prevenir posibles fallos o errores que se puedan cometer. De esta manera, se actuará de una manera eficaz y eficiente.

La planificación consta de varios puntos:

- Determinación de los objetivos.
- Análisis del área de trabajo.
- Técnicas de medición y equipamiento.
- Gestión de los datos.

El elemento objeto de estudio es un motor de un automóvil de la marca Kubota. Éste se encuentra asentado sobre dos apoyos de madera, que confieren estabilidad al conjunto. Con la finalidad de facilitar su traslado hasta el lugar de trabajo y poseer libertad de movimientos en sentido horizontal, se ha colocado sobre una estructura móvil. Para esta labor se ha contado con la colaboración de personal del grupo de investigación, debido a la imposibilidad de una única persona de moverlo y trasladarlo como consecuencia de su elevado peso.

Para evitar distintas exposiciones frente a la luz, las cuales podrían generar problemas durante el procesado, se opta por realizar la captura de datos dentro de una edificación cerrada. El lugar escogido para ello es el departamento de TIDOP, situado en la Escuela Politécnica Superior de Ávila. De esta forma, quedan establecidas unas condiciones de iluminación interior que serán invariables. En el supuesto de que sea necesario efectuar tomas en varios días, no se apreciarán las diferencias entre ellas.

En otro orden, y también como preparativo previo, se colocarán sobre los apoyos de madera dos patrones o testigos métricos. Estos elementos servirán de puntos de referencia entre las distintas capturas. Igualmente, al estar graduadas, se aprovecharán para acotar las dimensiones y poder comparar las reales con las conseguidas tras el procesado.

Antes de llevar a cabo el escaneo, hay que prever cuántas capturas serán necesarias para conseguir todos los datos que se utilizarán posteriormente. No deben quedar zonas ocultas o en sombra, puesto que serían regiones con información insuficiente. Del mismo modo, hay que evitar la redundancia de los datos, no realizando capturas desde el mismo punto de vista, ya que esto generaría mayor confusión de la información y un trabajo más laborioso a la hora de limpiar y alinear las distintas tomas.

3.1.2 Toma de datos Kinect 2

Para la adquisición de datos con el gaming sensor han sido necesarios los siguientes elementos:

- Kinect 2, con sus respectivas conexiones (corriente eléctrica y equipo de almacenamiento de datos).
- Tablet Levono Yoga 2 para el almacenamiento de datos, incluida la conexión a la corriente eléctrica.



Figura 4. Toma de datos Kinect 2

Para la captura de datos de la Kinect, el dispositivo encargado de realizarla, en este caso la tablet, ha de estar provisto del sistema operativo Windows 8 y poseer una conexión de entrada USB 3.0. Asimismo, son necesarios los drivers k4w o Kinect for Windows y un software que haga uso de ellos.

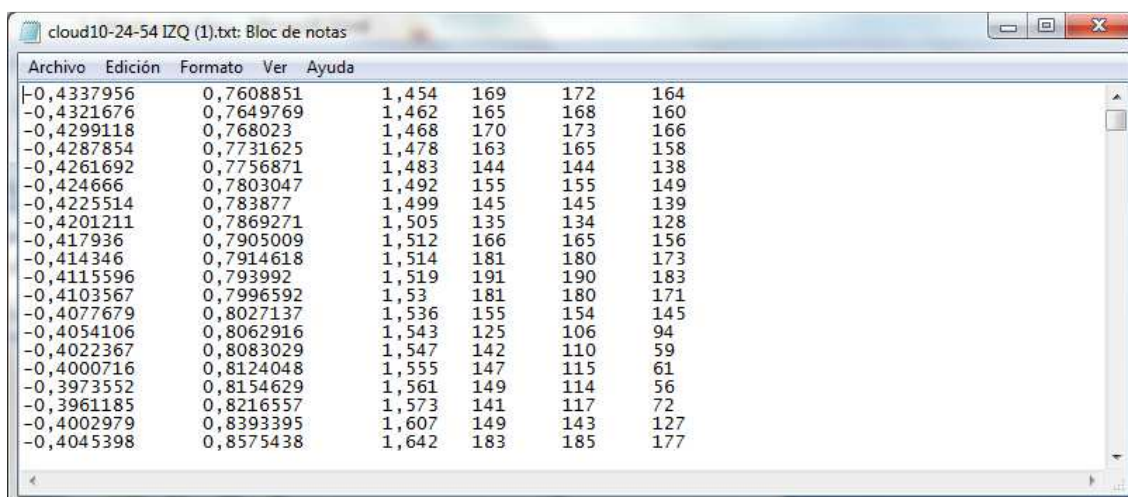
El equipo de almacenamiento se ha colocado sobre una mesa próxima al motor, puesto que la longitud de los cables, tanto de alimentación como de conexión con la

Kinect 2, tiene sus limitaciones. Unida a la tablet, y con una libertad de movimientos dentro del radio de acción (limitación longitud cable), se encuentra la Kinect 2. Esta autonomía nos permite realizar capturas desde distintos ángulos y posiciones.

La propia estructura móvil, sobre la cual se asentó el motor para su traslado hasta el lugar de estudio, ha servido de base también durante la adquisición de datos. La libertad de movimientos que posee facilita los desplazamientos y giros del objeto analizado. Gracias a ello, será posible tomar datos en un ángulo de 360º alrededor de él.

Una vez situados todos los elementos en el lugar adecuado se inicia la operación. El procedimiento que se lleva a cabo es el siguiente: mientras con una mano se orienta la Kinect en la dirección correspondiente, con la otra se realizan las capturas en la tablet. El método escogido es el de la toma en anillo. Dada la longitud de los cables, resulta imposible llevar a cabo toda la toma en anillo alrededor de los 360º manteniendo fijo el motor. Gracias a la plataforma móvil sobre la cual se asienta, éste será girado para poder continuar con la captura.

Un aspecto muy importante a tener en cuenta es la distancia comprendida entre la Kinect y el objeto de captura. Una distancia menor a la debida impediría obtener información y con una superior, se perdería. El rango de distancias admitidas que establece Microsoft para la captura de datos óptimos con la Kinect 2 se encuentra comprendido entre 0,5 y 4,5 m. Por lo que se opta por realizar las tomas con una separación aproximada de 1 metro. A lo largo de todas las capturas se procura mantener la misma distancia entre el motor y el sensor.



The image shows a screenshot of a Notepad window titled "cloud10-24-54 IZQ (1).txt: Bloc de notas". The window contains a table of data with 7 columns and 25 rows. The data is as follows:

| Archivo | Edición | Formato | Ver | Ayuda | | |
|------------|-----------|---------|-----|-------|-----|--|
| -0,4337956 | 0,7608851 | 1,454 | 169 | 172 | 164 | |
| -0,4321676 | 0,7649769 | 1,462 | 165 | 168 | 160 | |
| -0,4299118 | 0,768023 | 1,468 | 170 | 173 | 166 | |
| -0,4287854 | 0,7731625 | 1,478 | 163 | 165 | 158 | |
| -0,4261692 | 0,7756871 | 1,483 | 144 | 144 | 138 | |
| -0,424666 | 0,7803047 | 1,492 | 155 | 155 | 149 | |
| -0,4225514 | 0,783877 | 1,499 | 145 | 145 | 139 | |
| -0,4201211 | 0,7869271 | 1,505 | 135 | 134 | 128 | |
| -0,417936 | 0,7905009 | 1,512 | 166 | 165 | 156 | |
| -0,414346 | 0,7914618 | 1,514 | 181 | 180 | 173 | |
| -0,4115596 | 0,793992 | 1,519 | 191 | 190 | 183 | |
| -0,4103567 | 0,7996592 | 1,53 | 181 | 180 | 171 | |
| -0,4077679 | 0,8027137 | 1,536 | 155 | 154 | 145 | |
| -0,4054106 | 0,8062916 | 1,543 | 125 | 106 | 94 | |
| -0,4022367 | 0,8083029 | 1,547 | 142 | 110 | 59 | |
| -0,4000716 | 0,8124048 | 1,555 | 147 | 115 | 61 | |
| -0,3973552 | 0,8154629 | 1,561 | 149 | 114 | 56 | |
| -0,3961185 | 0,8216557 | 1,573 | 141 | 117 | 72 | |
| -0,4002979 | 0,8393395 | 1,607 | 149 | 143 | 127 | |
| -0,4045398 | 0,8575438 | 1,642 | 183 | 185 | 177 | |

Figura 5. Archivo ASCII

Los archivos que se obtienen en cada una de las capturas son una imagen .PNG y un fichero .ASC. En el fichero ASCII están contenidas las coordenadas XYZ de los puntos así como el color en función de los distintos canales RGB. Las coordenadas de los puntos capturados por la Kinect se encuentran en metros. Hay que tener especial atención a las unidades, ya que otros medios pueden presentar otras distintas.

Tras el comienzo del análisis de los datos registrados el primer día, se concluyó que los resultados obtenidos eran insuficientes en determinadas regiones, por lo que fue necesario desarrollar una segunda sesión de toma de datos. Ésta se centró en captar aquellas zonas que habían quedado ocultas o con información deficiente.



Figura 6. Imagen capturada por el sensor Kinect 2

El número de capturas realizadas durante la primera sesión fue de 17, dos tomas en anillo de 8 cada una, a diferentes alturas y una de la planta. Inicialmente estas capturas generaron un total de 2.881.347 puntos. Hay que notar que las tomas están sin tratar ni optimizar, por lo que el número final de puntos se verá reducido notablemente.

Durante la segunda sesión se han hecho un total de 18 capturas, esta vez concentrándose en las zonas clave. El número de puntos generados en esta ocasión asciende a 3.085.490. Gran parte de la información recogida en ellas se solapa con la toma anterior, por lo que el trabajo de limpieza de ruido y redundancia de puntos ha sido mucho mayor.

El tiempo estimado para la realización de la toma de datos con la Kinect 2, incluyendo las dos sesiones que fueron necesarias, oscila alrededor de las 3 horas. En este período se engloba tanto el montaje del equipo, como del motor en la plataforma correspondiente, y por supuesto las diversas capturas.

3.1.3 Toma de datos brazo robótico

El equipo necesario para realizar esta fase está conformado por:

- Brazo robótico portátil (Romer Absolute Arm 7325 SI), incluyendo los distintos sistemas de sujeción y las conexiones a la corriente eléctrica y ordenador de captura.
- Ordenador de captura (Workstation z400).

Tras la instalación de los diversos componentes del brazo robótico, y antes de comenzar a trabajar con él, es necesario fijar su posición, para poder así establecer un sistema de referencia. Este emplazamiento ha de ser el adecuado para poder alcanzar todas las regiones objeto de estudio del motor.

A diferencia de la captura con la Kinect, en la cual se toman datos del conjunto completo del motor, en el caso del brazo robótico sólo se ha llevado a cabo en una de las caras. La región escogida pertenece a una de las zonas de menor complejidad en relación al número de elementos y formas descritas. La razón de esta decisión es poder analizar esa misma zona desde ambos medios de captura, ya que los resultados del gaming sensor en las regiones complicadas no serán óptimos.



Figura 7. Escaneado brazo robótico

El equipo del Romer Absolute Arm cuenta con varios elementos de captura. Por un lado se encuentran las sondas de contacto y por otro el escáner láser integrado. En función del trabajo que se vaya a desarrollar se empleará una opción u otra. A pesar de que la precisión de las sondas de contacto es mayor que la del láser escáner, su tiempo de captura es elevado, debido a que hay que definir geoméricamente cada primitiva geométrica presente en el elemento objeto de estudio. Debido a la complejidad de los elementos del motor se excluye esta metodología.

Antes de comenzar el escaneo hay que tener en cuenta unos parámetros previos. Por un lado el filtrado. Esto es la longitud máxima del borde o arista y el paso de interpolación (ancho de malla). En función del volumen de la pieza se escogerá la resolución de la malla más adecuada. Al establecer el ancho de malla determinado, hay que controlar la velocidad con la que se trabaja, puesto que si la velocidad es alta, aparecerán huecos, y si, por el contrario, es demasiado lenta, se filtrará información redundante.

Otro de los aspectos a tener en cuenta es la exposición de la superficie para conseguir una buena captura de puntos. Hay que distinguir el parámetro interno del nivel de adquisición de información y, el porcentaje de intensidad de la señal láser.

La orientación de la línea de barrido láser respecto a los bordes y chaflanes ha de ser perpendicular a ellos. De no ser así, la calidad de los escaneos se vería afectada. Cada una de las pasadas que se realiza ha de ser visible desde un único punto de vista, en caso contrario, la información se perdería, las normales se invertirían y desviarían y los puntos se enmascararían. En el caso de los bordes, se recomienda que su captura sea a 45º de inclinación, ya que de esta manera se podrán ensamblar con las superficies.

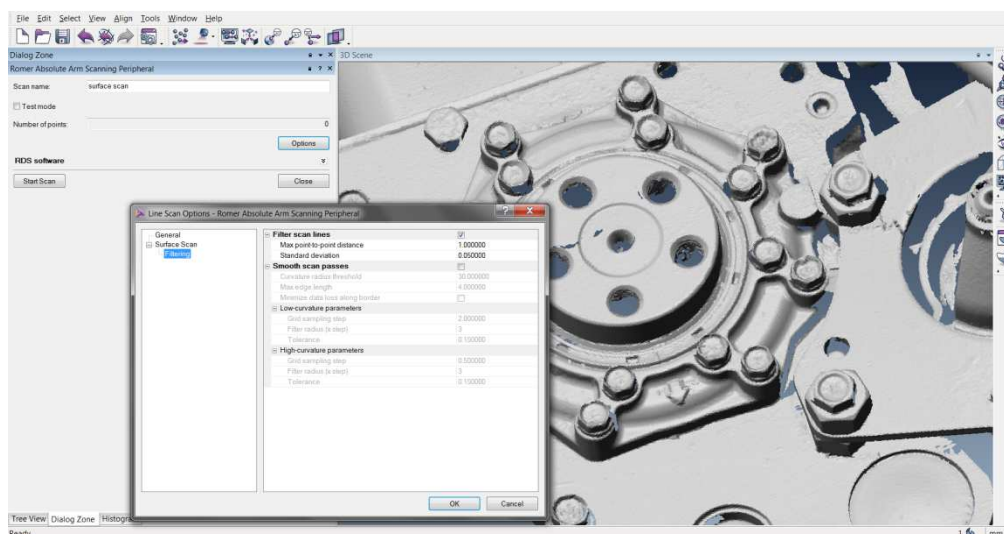


Figura 8. Captura y configuración

Los escaneos deben llevarse a cabo en una dirección con una única pasada. Con esto conseguimos evitar la redundancia de información y el ruido externo. Del mismo modo, se mejora la posterior extracción de los elementos característicos. La velocidad de escaneo está controlada por el movimiento del propio usuario. Debe ser lo más constante posible para evitar una mala definición, pérdida de información y la posible formación de huecos en los datos. Hay que evitar tanto las pasadas muy cortas, porque supondrían numerosos ensamblajes, como las que son demasiado largas, puesto que generarían fatiga en el operario.

El protocolo de captura ha consistido en realizar múltiples pasadas, hasta un total de 53. A lo largo de ellas se mantiene la distancia al objeto y la velocidad ha sido lo más constante posible, puesto que ésta afecta al número de puntos capturados. Al trabajar únicamente en una cara, no ha sido necesario mover el brazo, por lo que todas las pasadas están referenciadas en el mismo sistema.

Inicialmente se capturaron 1.395.786 puntos. Tras la posterior limpieza de ruido y optimización del resultado, el número de puntos se verá reducido. El software de captura empleado ha sido Polyworks Align V12.1, el cual se utilizará también durante el procesado.

Uno de los aspectos que se ha de tener en cuenta son las unidades con las que se trabaja durante la captura del objeto. Este interés tiene especial importancia durante la posterior comparación, en la que se confrontarán los datos conseguidos por el gaming sensor frente a los del brazo robótico. Mientras que la Kinect 2 desarrolla la toma de datos en metros, el brazo robótico lo hace en milímetros. Por tanto, una de las dos nubes o poligonales deberán ver modificadas sus unidades.

El tiempo de trabajo estimado tanto para la captura como para el procesamiento de los datos del brazo robótico es de unas 3 horas. En este período se incluyen todas las etapas: el montaje y la retirada del equipo, captura, limpieza, alineación y generación del modelo final.

En la toma de datos con el brazo robótico se ha contado con la colaboración de Pablo Rodríguez González, el cual ha mostrado su ayuda en todo momento y al que quiero mostrar mi agradecimiento.

3.2 Trabajo de oficina

3.2.1 Procesamiento datos Kinect 2

El procesamiento de los datos capturados por la Kinect 2 se ha desarrollado en su conjunto con un ordenador portátil Toshiba Satellite L750/L755 (Intel Core i7, tarjeta gráfica Nvidia Geforce y 4 Gb de RAM). Consta de varias fases diferenciadas:

- Limpieza de ruido.
- Alineamiento de nubes.
- Generación de malla.
- Creación modelo 3D.

Una vez finalizadas todas estas fases, y tras las diversas optimizaciones, se procederá a la comparación de las nubes de puntos o modelos generados frente a los obtenidos por el procesamiento de los datos capturados por el brazo robótico.

Dentro del procesamiento de datos obtenidos por la Kinect 2, hay que distinguir distintos tiempos de desarrollo, en función de cada fase. Por un lado, la limpieza del ruido y la alineación de las nubes de puntos se englobarán en una única tarea, a efectos de estimación de tiempo, ya que se trata de dos acciones complementarias que se irán intercalando progresivamente. Ésta fase, junto con la creación del modelo 3D, serán las dos que mayor duración poseen. En cambio, la generación de la malla previa al modelo tridimensional resultará prácticamente instantánea, dependiendo únicamente de la potencia del ordenador con el que se esté trabajando.

Por ello, en cada una de las etapas se comentará el intervalo de tiempo aproximado que se ha sido necesario en ellas.

A partir de los archivos conseguidos durante la captura, es decir, el archivo ASCII con las coordenadas XYZ y los colores RGB de cada uno de los puntos, y las imágenes en formato .png comenzará el procesamiento de los mismos.

Con ayuda de las imágenes .png podemos distinguir la vista capturada por la Kinect, lo que nos permite hacernos una idea de la nube de puntos previamente a su apertura.

Al tratarse de múltiples archivos, cuya únicamente diferencia en el nombre está en la hora de captura, se decide asignar a cada una de ellas un nombre que permita distinguirlas. Éste está referido a la posición respecto al motor.

La **apertura inicial y la limpieza del ruido** se llevarán a cabo con el programa CLOUDCOMPARE V2. Este software es capaz de abrir múltiples formatos, entre ellos el ASCII.

La ventana inicial que aparece previamente a la apertura del archivo, nos indica los elementos que se van a importar. Cada una de las columnas se corresponde con una coordenada (X, Y, Z) o un color (Rojo, Verde, Azul). Esto nos permite conocer las unidades en las cuales estamos trabajando. Hay que prestar atención a estas unidades porque la Kinect trabaja en metros y el brazo en milímetros.

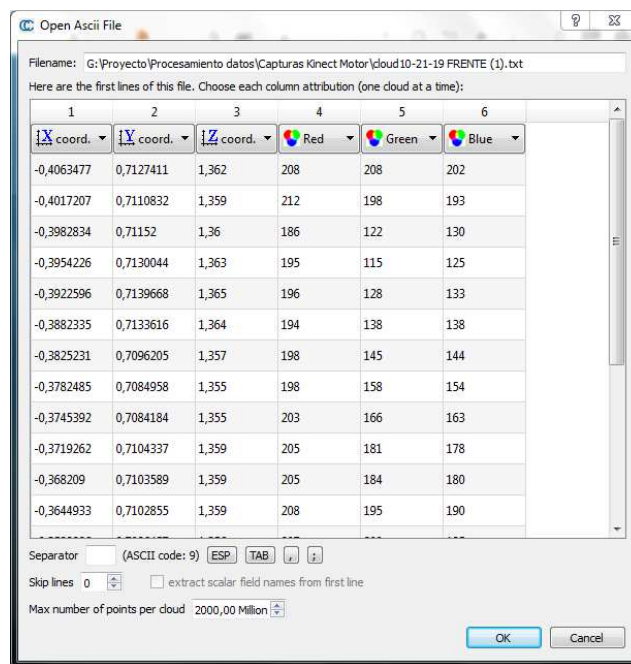


Figura 9. Entrada datos CloudCompare

Las nubes de puntos generadas tras la captura poseen una gran cantidad de ruido que hay limpiar. Es imprescindible eliminar todos aquellos puntos innecesarios, tanto los que no aportan ningún tipo de información al objeto de este estudio como los redundantes.

Conforme se pierde la perpendicularidad al dispositivo de captura, o aumenta la distancia, los puntos capturados pierden su veracidad tridimensional debido a la degradación de la señal de retorno y al aumento de error con la distancia. Es por ello por lo que hay que tener especial cuidado en la elección de los puntos correctos, lo cual incrementa el trabajo desarrollado.

Aunque CloudCompare permite abrir varios archivos simultáneamente, para trabajar con cada uno de ellos de manera más sencilla ocultaremos el resto.

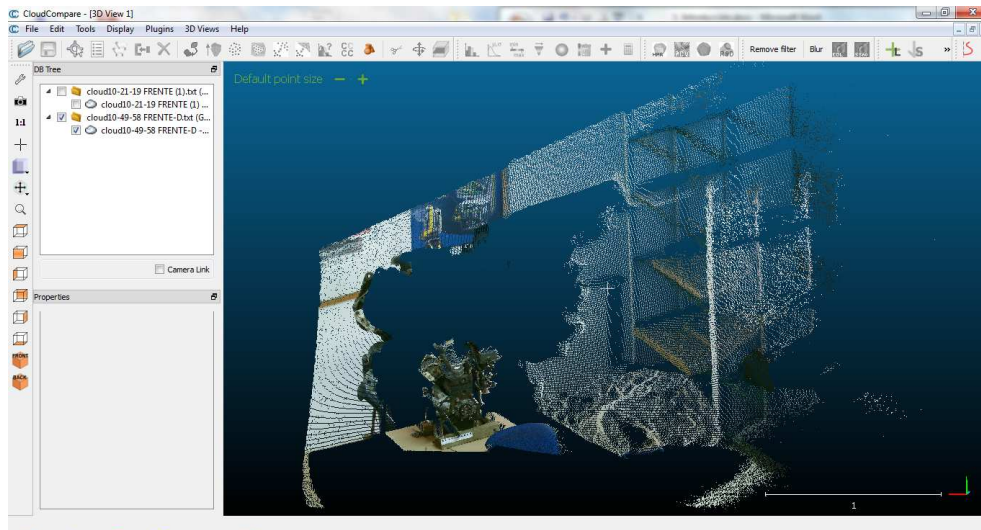


Figura 10. Nube de puntos con ruido

La supresión de las regiones de puntos que no son de utilidad para el objetivo final de la recreación en 3D, sigue las indicaciones que se muestran a continuación. En primer lugar se selecciona la nube de puntos que se desea modificar y se activa Segment. Esta herramienta tiene dos opciones de trabajo: se puede recortar la zona con la que nos queremos quedar, o bien, eliminar los puntos que deseamos desechar. Ambos planteamientos han sido manejados en función de los requerimientos en cada momento.

Así mismo, las nubes se pueden dividir y reagrupar nuevamente en función de las necesidades, a través de “Merge multiple clouds”. Tiene especial interés en aquellas situaciones en las que el trabajo con la nube de puntos global se encuentra avanzado, y tan sólo queremos trabajar en una zona reducida para mejorarla. Esto ocurre principalmente tras la segunda fase de toma de datos, en la cual sólo se van a añadir datos o puntos en localizaciones muy concretas.

Tras la laboriosa tarea de eliminar todo lo que es prescindible, el número de puntos de cada nube se ve reducido notablemente, simplificando así el peso del archivo. A continuación, se procede al **alineamiento** de las distintas **nubes**.

CloudCompare ofrece la posibilidad de alinear nubes o mallas basándose en la elección de pares de puntos homólogos. Permite implementar tantos pares como se crea conveniente. Sin embargo, los errores cometidos en la selección de los puntos, dada la imposibilidad de realizar esta acción con una precisión exacta por parte del usuario, no pueden verse minimizados.

Este fenómeno nos hace buscar otra solución que ofrezca unos resultados mejores. Por ello, y a pesar de que Cloud Compare posee una herramienta de alineación entre dos nubes, se escoge desarrollar esta fase con otro software como es REALWORKS SURVEY ADVANCED 6.4, cuya principal diferencia a la hora de alinear, es que permite aplicar un refinamiento tras la alineación inicial.

Antes de llevar los archivos al nuevo programa, hay que guardarlos en un formato compatible, como es ASCII. Se exigirá la máxima precisión admitida al exportar y como separador entre columnas se elige el espacio.

Entre las múltiples aplicaciones de edición de nubes de puntos y creación de mallas, RealWorks permite alinear y agrupar distintas nubes. Este alineamiento se puede realizar de dos formas, utilizando tarjetas de puntería o basándonos en superficies. En este caso, nos valemos de los puntos singulares de la propia geometría del objeto, y seremos ayudados por los patrones que se colocaron inicialmente sobre los apoyos de madera.

La orientación con puntos naturales se lleva a cabo con Cloud-Based Registration Tool, que no utiliza puntos artificiales (dianas de puntería), sino puntos pertenecientes a la nube.

Desde el apartado de Registration, se selecciona el proyecto y el método de alineamiento comentado anteriormente. Este sistema utiliza tres pares de puntos homólogos, elegidos por el usuario, que sirven de aproximaciones iniciales. Una vez que las nubes se encuentran en un sistema de coordenadas similar (siempre existe un error de precisión originado por el operario), se procede a la optimización y refinamiento del conjunto, todo ello siguiendo el **algoritmo ICP (Iterative Closest Point)**.

Este algoritmo, que fue creado originalmente por Bels y McKay. Tiene como punto de partida las dos nubes de puntos que se pretenden alinear. Se trata de un proceso iterativo en el que el objetivo es encontrar una matriz de rotación y traslación que permita a una de las nubes acercarse lo máximo posible a la de referencia. En cada iteración, primero se calculan las correspondencias de puntos usando el criterio del punto más cercano y, posteriormente, se calcula la matriz de rotación y traslación que minimiza la distancia entre los dos conjuntos de puntos, es decir, el error cuadrático medio (Bels, P.J., McKay, N., 1992).

Se caracteriza por el escaso coste computacional en cada una de sus iteraciones, lo que se traduce en velocidad de cálculo. Sin embargo, presenta dos grandes inconvenientes. Por un lado, una imagen debe ser un subconjunto de la otra, ya que el algoritmo es muy sensible a la presencia de outliers. Por otro, es necesario poseer una buena aproximación inicial. En caso contrario, el alineamiento no sería correcto, con

numerosos fallos y requeriría mucho más tiempo de ejecución al tener que probar distintas posiciones iniciales.

Con cada alineamiento, el error acumulado, debido tanto a la falta de precisión por parte del usuario en la elección de los puntos como al propio del aparato de captura, va aumentando. Para minimizar este error, en la medida de lo posible, se realiza el alineamiento por partes en ambos sentidos de giro.

De manera complementaria a las alineaciones en RealWorks, estos archivos han sido exportados simultáneamente a CloudCompare con el fin de reducir todo el ruido posible en las agrupaciones parciales. Disminuyendo de este modo el número de puntos y, por tanto, su peso, ofreciendo así una vista más limpia.

Los errores medios generados en las diferentes alineaciones después de aplicar el algoritmo ICP oscilan entre 1 – 3,5 mm dependiendo del caso.

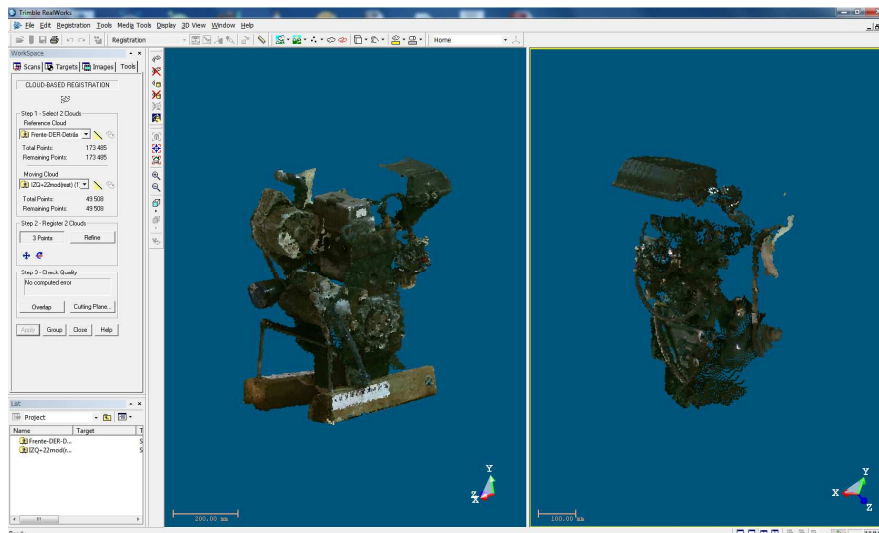


Figura 11. Alineamiento nubes de puntos 1ª fase

Tras el alineamiento y refinamiento del conjunto de los datos obtenidos en la primera toma, se generó una nube final que alcanzaba los 339.725 puntos, frente a los 2.881.347 capturados. Sin embargo, esta nube presentaba ciertas ocultaciones y zonas de sombra, en las cuales no se había podido recoger la información. Esto derivó en la necesidad de una nueva sesión de captura con la Kinect, para tratar de solventar estas carencias.

De esta manera, el segundo día de trabajo de campo, se efectuaron 18 capturas de las zonas requeridas, acumulando un total de 3.065.490 puntos. Siguiendo el mismo proceso que en la primera fase, se limpiaron las nubes, se alinearon y refinaron, y

nuevamente se volvieron a limpiar, acoplado únicamente las regiones deseadas. En este caso, el proceso del alineamiento no ha llevado el mismo orden que en la primera fase, donde se iban incorporando las nuevas nubes de puntos siguiendo un orden de giro. Al tratarse únicamente de regiones aisladas, no ha sido imperioso desarrollar tal orden, si no que se han ido incorporando únicamente en los puntos clave. Finalmente se obtuvo una nube formada por 577.691 puntos.

Se estima que el período de tiempo empleado en estas dos fases de limpieza y alineamiento de nubes es de unos 15 días con una media de 4 cada uno de ellos. Esto hace un total aproximado de 60 horas activas. Esto es debido a que la tarea del alineamiento requiere, en muchas ocasiones, numerosos intentos en el momento de selección de los pares de puntos, como consecuencia del excesivo ruido, a pesar de que éste haya sido limpiado al máximo posible.

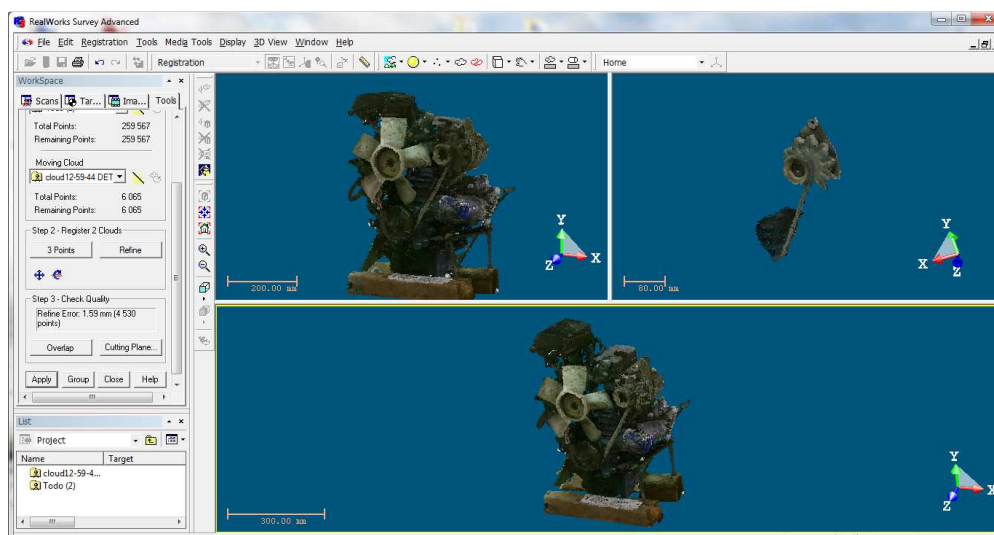


Figura 12. Alineamiento nube de puntos 2ª fase

Una vez que se ha terminado el procesamiento de la nube de puntos, se procede a la **creación de la malla**, la cual nos habilitará posteriormente a la reconstrucción en 3D del modelo. Para esta labor nos valemos de la herramienta de RealWorks de generación de mallas.

Dada la gran cantidad de puntos que poseía la nube, el equipo con el que se ha desarrollado el trabajo no era capaz de procesar esta acción. Con el fin de agilizar el procesamiento se ha diezmado la nube de puntos. El inconveniente que presenta es que se pierde exactitud. Pero tras la valoración entre calidad y tiempo de ejecución, se decide escoger esta opción.

El proceso de diezmo de la nube se puede hacer directamente en el momento de creación de la malla (Mesh Creation Tool) a través de Sampling Tool. Tras realizar varias pruebas, finalmente se decide diezmarla al 10%, quedando por tanto, un total de 57.769 puntos. Antes de conseguir la malla definitiva, ésta se puede editar eliminando el mayor número de impurezas posible con Mesh Editing Tool. El proceso de refinamiento, el cual cuenta con tres fases (Refine > Smooth > Remove peaks) se aplica de manera secuencial, existiendo la posibilidad de hacerlo varias veces. El número de iteraciones tampoco debe ser excesivo, puesto que supondría una elevada pérdida de información.

La malla será exportada en un formato compatible con el software de Geomagic que será el que nos permitirá generar finalmente el modelo 3D. El formato elegido ha sido .OBJ, el cual representa la geometría en 3D conociendo la posición de cada uno de los vértices y las caras de los polígonos definidos por los anteriores.

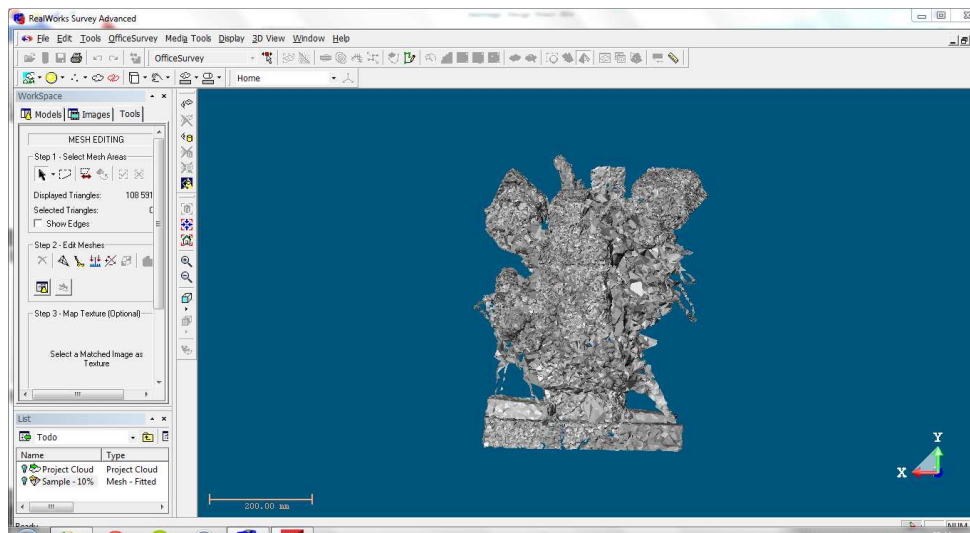


Figura 13. Malla 10% creada con RealWorks

Esta fase se podría haber realizado directamente con Geomagic, ya que posee un módulo de importación de nubes de puntos y creación de mallas. De hecho, se ejecutaron varias pruebas tanto con la nube de puntos al 100% como diezmada. Sin embargo, los resultados arrojados fueron inferiores respecto a los de RealWorks. El mecanismo de edición y mejora de la malla, aplicando las herramientas de Reducir ruido > Envolver > Reparar > Rellenar, generaban una gran pérdida de detalles, sobre todo en las regiones con oquedades. Por esta razón, se mantuvo la idea de continuar trabajando con la malla creada anteriormente con el otro software.

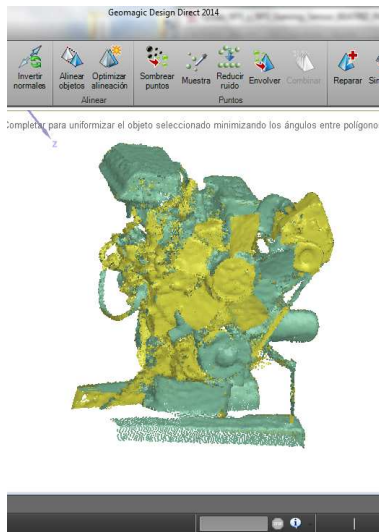


Figura 14. Reducir ruido

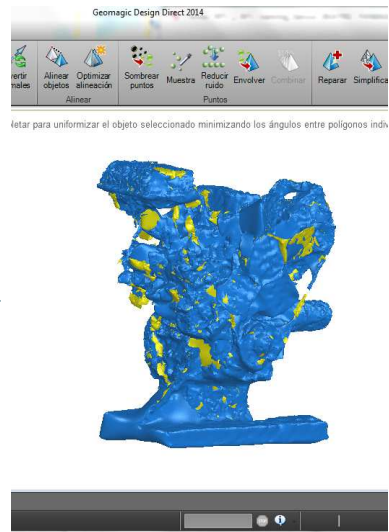


Figura 15. Envolver

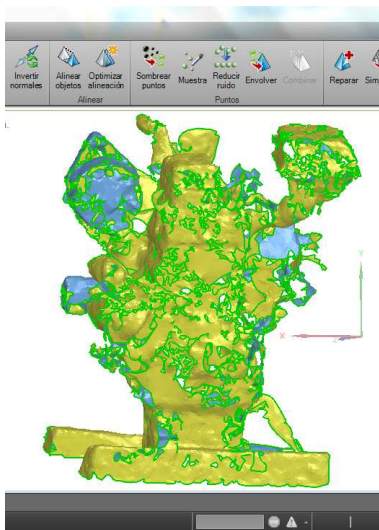


Figura 16. Rellenar

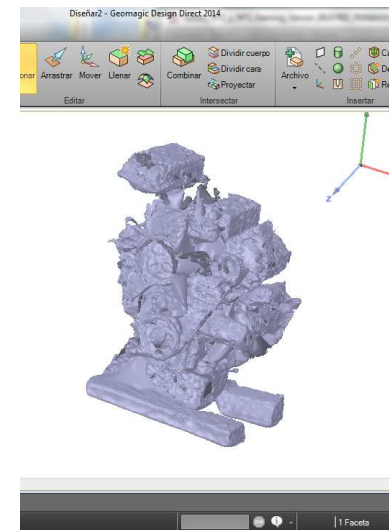


Figura 17. Malla creada por Geomagic

El intervalo de tiempo inevitable durante la generación de la malla depende por un lado de la cantidad de puntos que posea la nube, y por otro, de la potencia del ordenador con el que se esté llevando a cabo. En este caso, fueron necesarios varios intentos, ya que con el 100% de la malla era incapaz de procesarlo.

La creación de la malla final precisa tan sólo de un par de minutos. Sin embargo, al haberse realizado otras pruebas con distintas cantidades de puntos, se podría decir que el tiempo total empleado duró varias horas.

La última parte del procesamiento la constituye la reconstrucción del **modelo** del motor en **3D**. Para ello se emplea el software de GEOMAGIC DESIGN DIRECT 2014. Junto con la limpieza y alineamiento de las nubes de puntos, esta fase es una de las que mayor tiempo de trabajo ha supuesto en este proyecto. La extracción de los múltiples elementos que conforman el motor, así como el ensamblaje y conexión entre las distintas superficies, se traduce en una enorme dedicación y paciencia por parte diseñador.

La recreación de un modelo en 3D a partir de una malla engloba distintas acciones, todas ellas complementarias entre sí. El primer paso, antes de todo, lo constituye la importación del archivo .obj generado en RealWorks. Dado que la malla ha sido creada a partir de una nube de puntos diezmada al 10%, existen varias regiones que no se encuentran plenamente definidas, las cuales no será posible reconstruir.

A partir de la malla tridimensional se comenzarán a extraer las distintas superficies y elementos que componen el motor. El software de Geomagic ofrece varias formas de extracción: ajustar formato libre, plano o cilindro, entre otras. Estas que han sido nombradas son las que se han empleado durante la creación de los diversos componentes.

Durante el procedimiento de extracción, en primer lugar se selecciona una región a partir de la cual se generará una superficie que se ajusta lo máximo posible a la zona de la malla. Hay que tener en cuenta, que la malla está formada por triángulos, y en ocasiones, las distintas caras provocan que la superficie extraída no se ajuste plenamente a lo que deseamos, y por ello, sea necesario editarla.

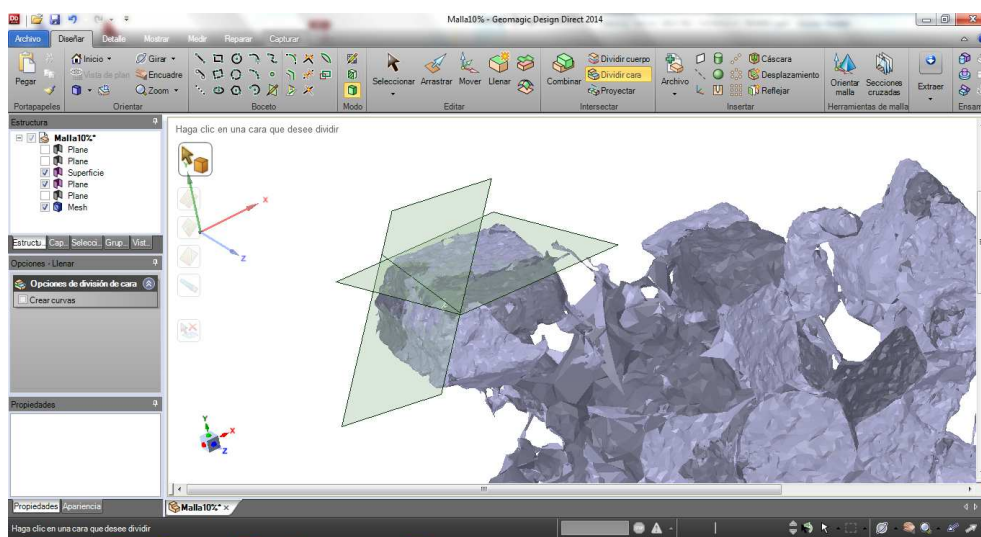


Figura 18. Dividir cara

En este aspecto, el programa presenta ciertas limitaciones y podría ser un punto de avance y desarrollo en el futuro. Las extracciones que realiza únicamente se ajustan a geometrías básicas, como planos, esferas o cilindros, a pesar de que ofrece una opción de ajuste de formato libre. Por tanto, sería de gran interés el desarrollo de algoritmos que permitiesen obtener cualquier tipo de geometría, que se encontrase plenamente definida, tan sólo con una acción, y no realizando este proceso de manera manual y engorrosa, como ha sido el caso del presente proyecto.

A pesar de estas restricciones, se ha tratado de obtener los diferentes elementos. El caso más particular de extracción ha sido la correspondiente al ventilador, puesto que no presentaba ninguna superficie plana o cilíndrica. Manualmente se ha ido tejiendo la geometría descrita por las aspas, recolocando cada uno de los puntos que componen la superficie generada por el ajuste de formato libre. Una vez adaptada la superficie a la malla, se procede a su extrusión y posterior creación de patrón circular.

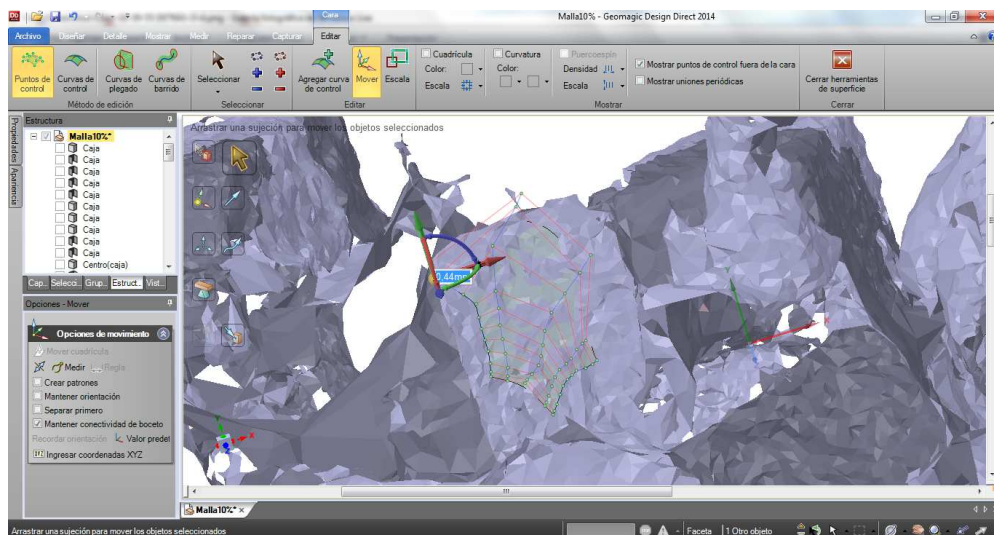


Figura 19. Creación ventilador

A lo largo de la reconstrucción han sido varias las herramientas utilizadas, en función de la geometría. Partiendo de las extracciones de superficies en los elementos mejor definidos, se sigue con la edición de las mismas. En algunos casos ha sido necesario girar, desplazar, recortar o extruirlas, pero también dividir caras o cuerpos, dibujar formas irregulares en el formato boceto, crear orificios o generar patrones circulares (ventilador y engranaje) o lineales.

Algunas de las herramientas más útiles a la hora de encajar las distintas piezas se localizan en el apartado de ensamblaje. Nos permiten orientar y alinear de manera apropiada los distintos componentes. En ocasiones, durante el diseño de los distintos fragmentos, es indispensable establecer unas condiciones de tangencia y orientación entre ellos, para la correcta creación del conjunto. Este proyecto únicamente está dedicado al análisis de los distintos equipos de captura de datos, pero en el caso de la ingeniería inversa en el campo industrial, resulta imperioso que todas las piezas encajen en su lugar correspondiente, pudiendo así funcionar el mecanismo.

Poco a poco se van creando y uniendo los distintos elementos que componen el motor, hasta alcanzar el modelo final. Aquellas partes complicadas, donde se confunden los datos obtenidos quedarán sin reconstruir. Estas regiones se concentran principalmente en una de las caras y en la planta, y se corresponden con las zonas donde se encuentran los tubos. Las finas tuberías y enrevesadas trayectorias que siguen, hacen que se pierda gran detalle tanto en la nube de puntos, como posteriormente en la generación de la malla. Es por ello por lo que no se dispone de la suficiente información para crear superficies y sólidos en estos sectores y quedará un vacío en los mismos.

Tras la reconstrucción de los principales cuerpos que componen el objeto estudiado, se aplican una serie de optimizaciones, con el fin de mejorar el resultado final. Dentro del propio diseño y edición se redondean varios bordes gracias a la herramienta de arrastrar. Fuera de éste, y en el apartado de reparar aplicaremos diversas funciones de corrección y ajuste:

- Solidificación de espacios
- Corrección de bordes inexactos, caras duplicadas y bordes extra.
- Corrección de curvas (espacios entre ellas, curvas pequeñas y ajuste de curvas).
- Ajuste de caras pequeños y fusión de caras.

Finalmente, tras crear un modelo en 3D del motor, lo más análogo posible a la malla que se ha tomado como referencia, se exporta este archivo en formato .STL, compatible con CloudCompare, para poder llevar a cabo la comparación del mismo con la nube de puntos obtenida por la Kinect 2 y el modelo 3D conseguido con el brazo robótico. STL es un formato de CAD que define la geometría de objetos 3D, excluyendo información como colores o texturas.

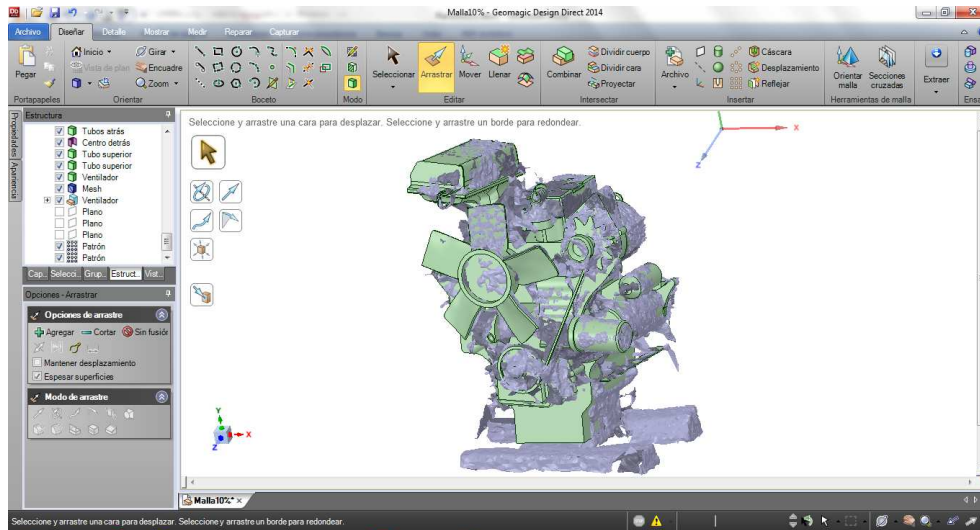


Figura 20. Reconstrucción 3D

Una de las partes más laboriosas de este trabajo ha sido la creación del modelo 3D del motor. En esta etapa han sido necesarios unos 18 días durante una media de 5 horas cada uno de ellos. Por tanto, el número total de horas aproximado se acerca a las 90.

3.2.2 Procesamiento datos brazo robótico

Durante el procesamiento de los datos obtenidos por el brazo robótico Romer Absolute Arm se ha empleado el mismo software que se utilizó en su momento para la captura, es decir, POLYWORKS ALIGN V12.1. Del mismo modo, se ha trabajado con una Workstation z400 (Intel Xeon w3550, 6 Gb de RAM y tarjeta gráfica Nvidia Quadro 4000 de 2 Gb).

A diferencia de los datos capturados por la Kinect 2, los cuales han tenido que pasar por distintos programas en cada una de sus fases (CloudCompare, RealWorks y Geomagic Design), para el caso de los datos conseguidos a través del brazo robótico, únicamente será preciso disponer de uno, Polyworks. Se trata de un software de metrología en 3D universal para el procesamiento y análisis de datos tridimensionales. Permite trabajar desde la captura de nubes de puntos, hasta el procesamiento y modelado de superficies, así como la comparación y el estudio frente a otras.

El procedimiento a seguir, es parecido al que se ha llevado a cabo con los datos obtenidos por la Kinect:

- Limpieza del ruido.
- Alineamiento y refinamiento.

- Creación modelo poligonal.

En el caso del brazo robótico sólo se ha realizado la captura de parte de una de las caras del motor, de tal modo, que el trabajo de procesamiento de los datos será inferior al que ha tenido lugar con los de la Kinect.

En primer lugar, y como paso inicial en cualquier procesamiento de puntos, es necesario llevar a cabo una **limpieza del ruido**. Durante la toma de datos se captura mucha información que no tiene utilidad. Puede ser debido al entorno o las condiciones de iluminación, pero la principal causa es el equipo empleado. En el caso de los brazos robóticos también influye el modo de captura. La velocidad de las pasadas y la distancia desde la cual se realizan son aspectos a tener en cuenta por su enorme influencia en este aspecto.

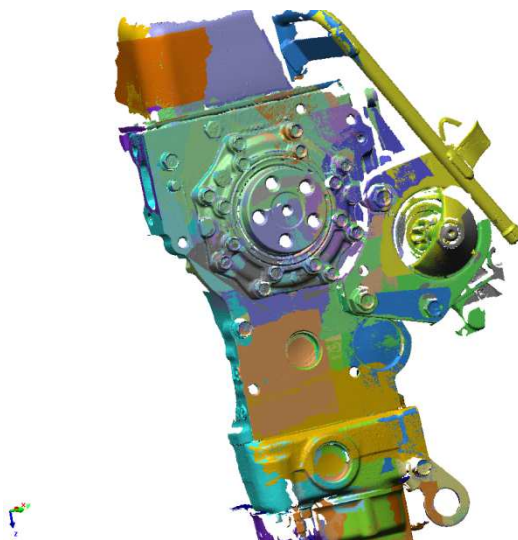


Figura 21. Color según pasada

Destaca la notable diferencia que existe entre el ruido generado en cada una de las tomas del gaming sensor frente al obtenido por el brazo robótico. Mientras que en el primero de ellos aparece gran parte de la sala donde se ha desarrollado la toma de datos, el segundo se limita al objeto de estudio. En él, la principal fuente de ruido procede de los apoyos de la mesa, los cuales han sido eliminados.

Una vez finalizada la fase de limpieza de ruido, el siguiente paso lo constituye la **alineación** de las distintas tomas. La localización tanto del brazo robótico como de la región que ha sido objeto de estudio ha permitido que todas las capturas se hayan realizado desde un mismo punto, es decir, no ha sido necesario mover el brazo. Como

consecuencia, todas las pasadas están referenciadas en el mismo sistema y, por tanto, alineadas, gracias a la tecnología del propio brazo para ubicar la posición del escáner láser.

Polyworks utiliza las áreas de solapamiento para realinear todas las pasadas. En primer lugar, realiza un análisis de desviación en la nube de puntos alineados. A partir de ahí, calcula automáticamente las promedio entre cada pasada y muestra el valor de la desviación con una superposición de colores. De esta manera, se pueden identificar las zonas que se encuentran fuera de la tolerancia admitida y ejecutar nuevas alineaciones para su mejora, en el caso de que fuesen necesarias.

Posteriormente, se optimiza el resultado con la tecnología “best-fit alignment”. Los parámetros que se tienen en cuenta durante esta alineación son:

- Distancia máxima (se establece una longitud determinada para buscar puntos homólogos).
- Submuestreo (filtrado para acelerar los cálculos).
- Iteraciones. Supone una alternativa al criterio de desviación estándar.

El alineamiento se lleva a cabo en varias etapas. Primeramente se realiza una aproximación inicial con submuestreo y después un refinamiento de la nube de puntos.

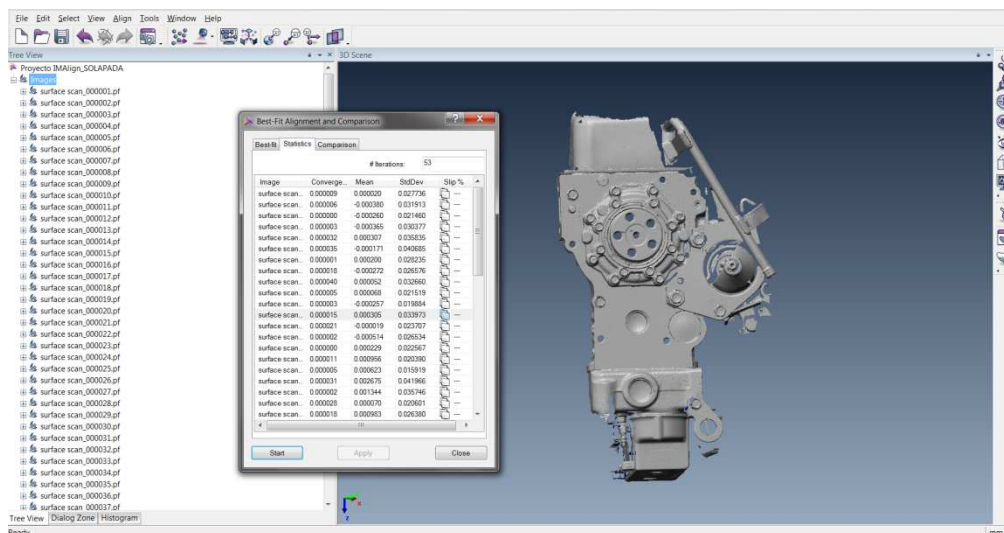


Figura 22. Best fit align

En el supuesto de que las pasadas fuesen demasiado cortas, no habría suficientes puntos para poder anclarlas. Para evitar el deslizamiento entre ellas, se fijan ocho

puntos de control para cada escaneo, los cuales restringen los grados de libertad. Después de aplicar best-fit alignment se vuelve a ejecutar un análisis de desviación con su correspondiente mapa de colores.

El siguiente paso del alineamiento y el refinamiento consiste en crear un modelo de nube de puntos de una sola capa, valiéndose de la tecnología “Reduce overlap”. De esta manera se reduce el número de puntos total, debido al solape de las múltiples pasadas. Finalmente, se optimizan los radios de curvatura a través de “Keep best line data”.

A raíz de todas estas mejoras, pasamos de los 1.395.786 puntos que se capturaron inicialmente a los 1.014.817 que quedaron tras las optimizaciones.

En el último lugar del procesamiento se encuentra la generación del **modelo poligonal** en formato .STL con el módulo IMMerge. Con el fin de poder comparar los modelos obtenidos por ambos equipos, es necesario convertir las nubes de puntos en superficies. Polyworks nos permite transformar los más de un millón de puntos en una malla triangulada. Utiliza cada uno de los puntos como nodos para los triángulos que construye. En este proceso se ha usado el nivel de suavizado Medio, con una distancia máxima de 0.5 mm.

Hay que recordar que las unidades con las que se están trabajando en este procesamiento son milímetros, frente a los metros con los que se ha obtenido el modelo en 3D del motor a partir de los datos capturados por la Kinect 2.

Antes de realizar las diversas comparaciones de errores que se llevarán a cabo, hemos de igualar las unidades en ambos archivos. No sólo tendrá lugar en el modelo poligonal final, sino que también a nivel de nube de puntos y malla, puesto que estas serán utilizadas del mismo en las diferentes confrontaciones.

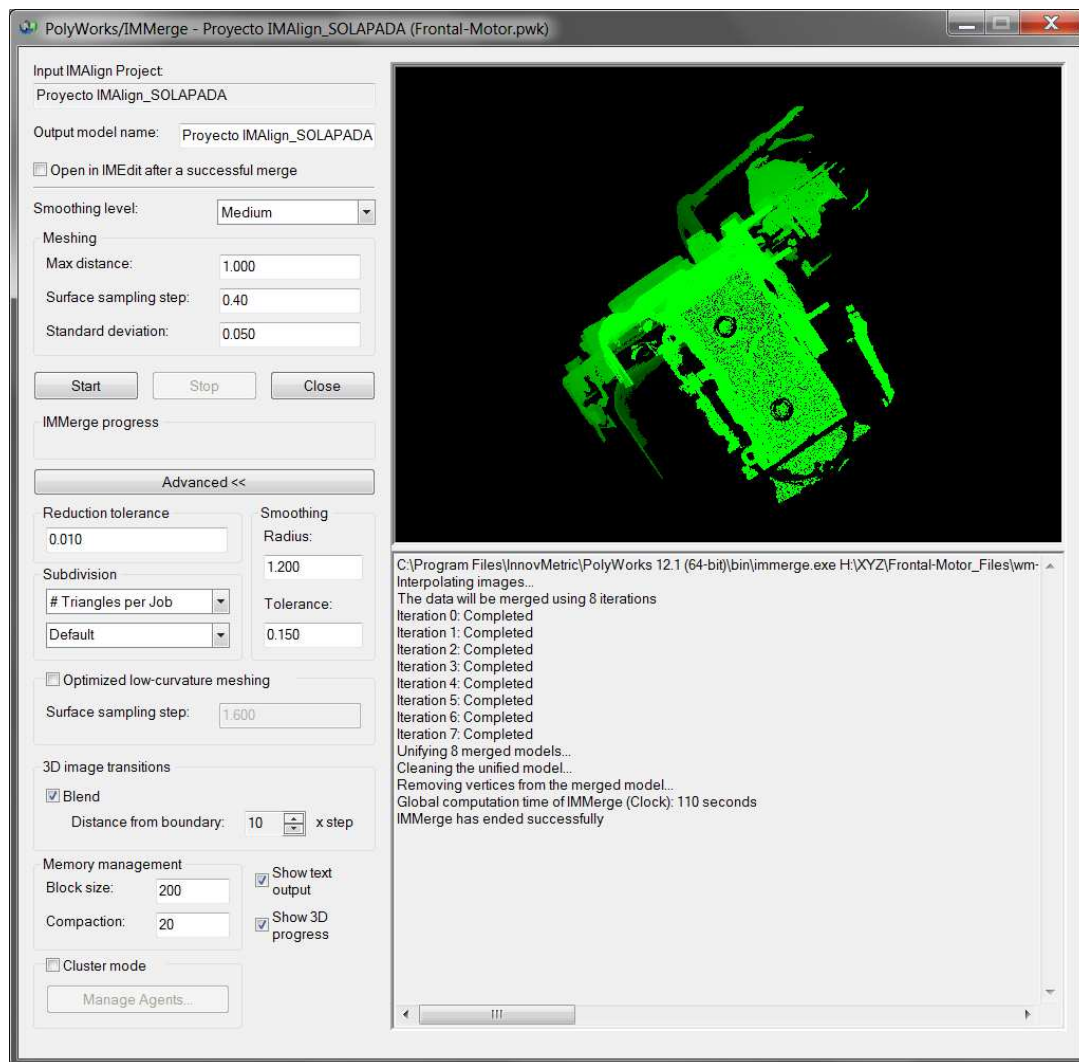


Figura 23. Generación modelo poligonal IMMerge

4. COMPARACIÓN EQUIPOS

El principal objetivo que se ha establecido en el planteamiento de este proyecto, ha sido la comparación de distintos procesos de ingeniería inversa aplicados en elementos de automoción. Los dos equipos que se han utilizado presentan inicialmente grandes desigualdades a nivel económico, y presumiblemente en los resultados finales.

Con el fin de demostrar estas disparidades y tener pruebas veraces de ello se procede a la equiparación entre los resultados obtenidos por ambos medios. Ésta se llevará a cabo a través del programa CloudCompare, que como su propio nombre indica, realiza comparaciones entre nubes de puntos. Es una herramienta que permite realizar estimaciones sobre las diferencias entre dos mediciones de un mismo objeto, pudiendo así determinar por ejemplo las deformaciones. Nos permite computar las distancias entre puntos de dos nubes aparentemente similares con el fin de poner en evidencia y cuantificar, a través de mapas de falso color, las posibles diferencias globales o particulares entre modelos.

Los resultados que se obtengan en dichas comparaciones serán equiparados a los valores esperados inicialmente, que serían aproximadamente de unos 4,6 mm, como se indica en las especificaciones y características de la Kinect 2.

En primer lugar, antes de importar los archivos a comparar, éstos deben haber sido guardados o exportados en las mismas unidades en los software de procesamiento correspondientes. Hay que recordar que los datos obtenidos por el gaming sensor se encontraban en metros, mientras que los del brazo robótico en milímetros. Es por ello, por lo que se decide guardar y convertir las nubes de puntos y mallas creadas por la Kinect 2 en milímetros.

Una vez que todos los datos se encuentran en las mismas unidades, será posible contrastarlos. Al trabajar con las nubes en milímetros, se entiende que los errores reflejados también estarán en milímetros.

CloudCompare permite hacer comparaciones entre dos nubes de puntos y entre nubes de puntos y mallas. Por ello, se llevarán a cabo las siguientes:

- Nube de puntos individual Kinect vs. Nube de puntos global Kinect.
- Nube de puntos Kinect vs. Malla modelado 3D.
- Malla brazo robótico vs. Malla modelado 3D.

- Nube de puntos Kinect vs. Malla brazo robótico.

Previamente a la comparativa entre las distintas nubes y mallas es imprescindible que estén referenciadas en un sistema de coordenadas común, para poder cotejar las distancias comprendidas entre ellas. A diferencia de las alineaciones entre las diferentes nubes de puntos, las cuales se llevaron a cabo con RealWorks, en este caso, todo el proceso de comparación, desde la alineación inicial se ejecutará con CLOUDCOMPARE.

Al tratarse de archivos cuyas coordenadas absolutas de referencia no son similares, es necesario realizar una alineación manual entre ellos. Este software de edición y equiparación de nubes, presenta un apartado de alineamiento. El proceso consiste en la elección de pares de puntos homólogos de los dos archivos con los que se está trabajando. Permite seleccionar tantos pares como sean necesarios, para mejorar así su alineación. Para poder computar las distancias entre los dos modelos, se aplica una transformación geométrica sólido-rígido, de tal forma que ambos pasen a ocupar un mismo espacio, a partir del cual poder medir y estimar las diferencias existentes. Hay que tener en cuenta que se pueden generar falsas diferencias por defectos en la alineación de ambos modelos, es decir, errores generados por el propio usuario en el momento de la elección de los puntos.

Estas alineaciones se llevan a cabo sin aplicar ningún tipo de ajuste de escala, puesto que se supone que ambos sensores son métricos.

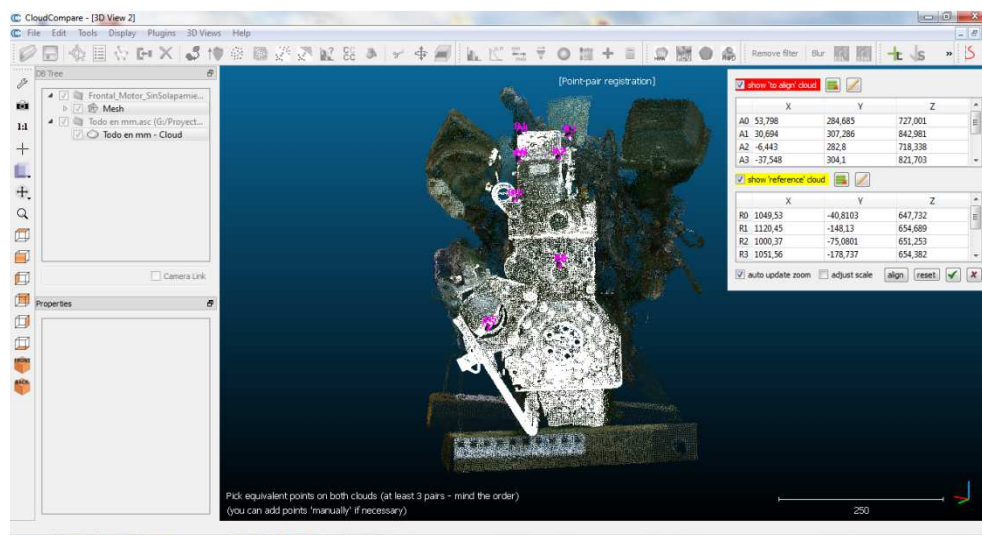


Figura 24. Alineación nubes

Tras el alineamiento el programa nos ofrece información acerca del error cuadrático medio (RMS) final, la matriz de transformación aplicada, así como del factor de escala en el caso de que haya sido necesario.

Una vez que los modelos han sido alineados, es recomendable recortarlos para que los siguientes análisis se realicen exclusivamente sobre las partes comunes de ambos, de otra manera se distorsionarían los resultados finales, pudiéndose hacer inviable la identificación de las variables objeto de estudio.

Unas consideraciones previas que hemos de tener antes de realizar la comparación son:

- Asegurarnos de que la extensión de la nube de referencia es mayor que la comparada. La nube de referencia siempre se superpone a la de comparación, situación que también se cumplen con las mallas.
- Tratar de obtener una nube de referencia lo mayor densa posible, puesto que influirá en la exactitud de los resultados.
- En función del ruido de la nube de referencia emplear un método u otro. En el caso de contener una gran cantidad de ruido se recomienda utilizar el ajuste por mínimos cuadrados, puesto que es más robusto. En cambio, si poseemos una nube de referencia limpia con alta curvatura, la mejor opción se corresponde con la función de altura cuadrática.

Para realizar la comparación entre dos nubes de puntos es necesario seleccionar ambos archivos, alinearlos y, posteriormente activar la herramienta **Compute cloud/cloud distance**. Una de las dos nubes será la que se tome como referencia, mientras que la otra será la comparada. Este orden se puede invertir y elegir la nube que deseemos como la de referencia.

El usuario puede ajustar la distancia máxima comprendida entre los modelos, así como fijar una serie de parámetros que determinan la velocidad y rapidez de comparación. Previamente al resultado final, se muestra una previsualización de las diferencias en un mapa de color basado en la distancia de Chamfer, algoritmo muy utilizado en la identificación de disparidades entre imágenes.

Tras calcular las distancia locales entre dos nubes de puntos densas, filtra las medidas de ruido para identificar las auténticas diferencias, y reconoce los elementos individuales que han variado entre los dos conjuntos de datos.

Después de la comparación inicial aparecen una serie de datos medios relativos a la mínima distancia, máxima distancia, distancia media, desviación estándar y máximo error relativo. Todo ellos se corresponden con unos resultados medios aproximados.

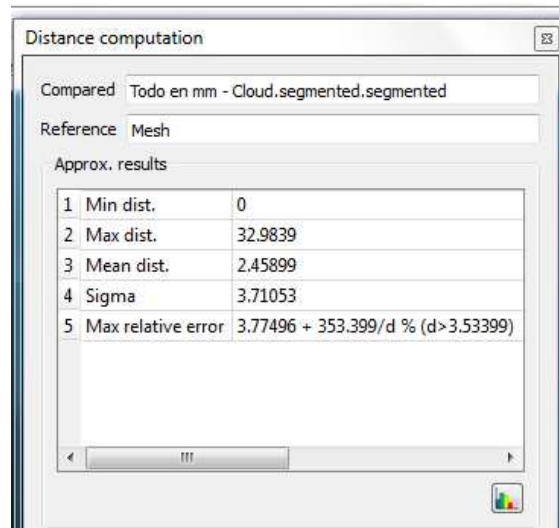


Figura 25. Ventana de resultados aproximados

Dentro de la comparativa entre nubes se pueden distinguir varios métodos, los cuales los encontramos en la ventana de Local Modeling. En función del método de cálculo empleado, se reflejarán una serie de diferencias. Los distintos métodos que ofrece son:

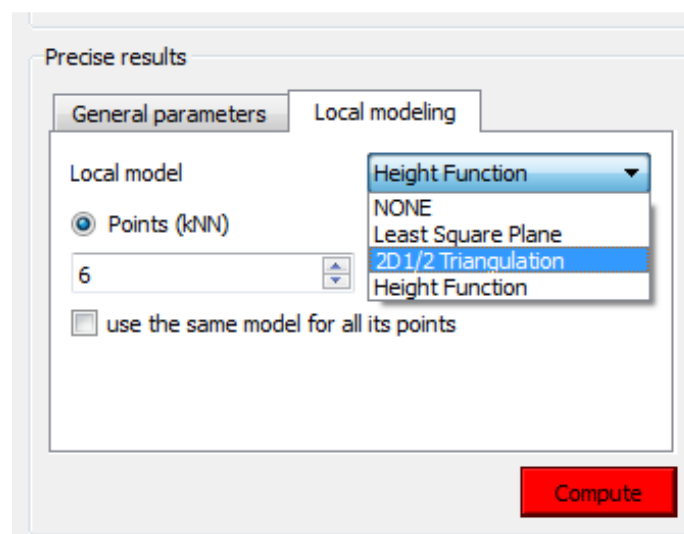


Figura 26. Tipos de modelización

- *NONE (Algoritmo del vecino más cercano):*

Es el método que viene por defecto al iniciar el programa. Se corresponde con la búsqueda del vecino más cercano. CloudCompare examina cuál es el punto más cercano en la nube de referencia y calcula su distancia euclídea.

El algoritmo de búsqueda del vecino más cercano se puede definir de la siguiente manera:

Dado un conjunto de puntos $P = \{p_1, \dots, p_n\}$ en un espacio X con función de distancia d , permitiendo algún preprocesamiento en P de manera eficiente, se plantean dos cuestiones:

- *Localizar el punto en P más cercano a $q \in X$.*
- *Dado un punto $q \in X$, y $r > 0$, regresar todos los puntos $p \in P$ que satisfagan $d(p,q) \leq r$.*

Este problema puede ser resuelto en tiempo $O(dN)$ para N dimensiones, sin embargo, se suelen emplear preprocesamientos para acelerar la búsqueda. Este algoritmo de discriminación no paramétrica está basado en estimaciones de funciones de densidad de probabilidad, estimándola directamente a posteriori. Por tanto, están caracterizados por la ausencia de hipótesis a priori sobre la distribución condicional del espacio de definición. Puesto que la base está en el cálculo de distancias, esta puede ser quizás su mayor inconveniente.

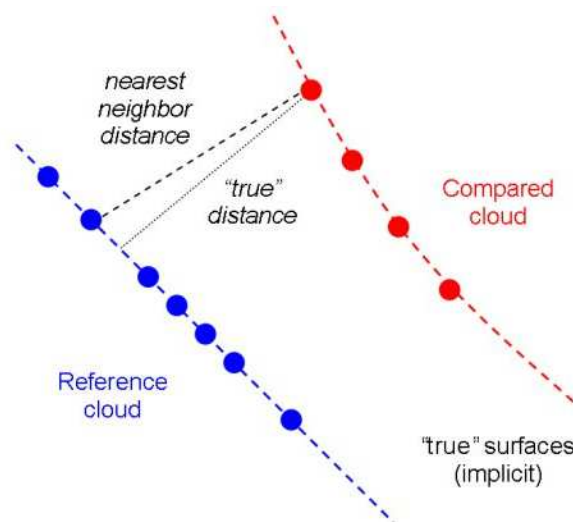


Figura 27. Búsqueda vecino más cercano

Si la nube de referencia es lo suficientemente densa, es decir, los puntos están cerca, la aproximación entre la nube comparada y la superficie subyacente representada por la nube de referencia, será aceptable. En el caso de que la nube no

sea lo suficientemente densa y los puntos se encuentren muy separados, el error obtenido será ligeramente superior al conseguido por otros métodos, como por ejemplo, la triangulación.

Este método es el más óptimo en el caso de poseer un modelo global limpio y adecuado. Cuando el programa ha determinado el punto más cercano en la nube de referencia, la idea es localizar el modelo de esta nube basándose en un ajuste matemático del punto más cercano y varios de sus vecinos. La distancia entre cada punto de la nube comparada hasta el más cercano en la de referencia, se sustituye por la distancia al modelo. Este proceso es más preciso y menos dependiente de la toma de muestras de la nube.

- LEAST SQUARE PLANE (Ajuste por mínimos cuadrados):

Es una de las técnicas más utilizadas para el ajuste de datos en diferentes campos como la fotogrametría, la geodesia o la topografía. Esto es debido a que proporciona un algoritmo computacional para obtener una respuesta única incluso en los casos más complicados.

No requiere ningún conocimiento a priori de la distribución de probabilidad. Tan sólo es necesario conocer la matriz peso o la matriz cofactor. Proporciona estimaciones de todas las variables del modelo, así como de sus respectivas matrices covarianza y cofactor.

Se suele llevar a cabo con ecuaciones lineales, puesto que en otro caso, la solución resultaría demasiado complicada. Cuando las ecuaciones del modelo son inicialmente no lineales, se utiliza algún método de linealización como el desarrollo en serie de Taylor.

El ajuste por mínimos cuadrados constituye el método más robusto en el caso de nubes de puntos con ruido y valores atípicos. Trata de crear una aproximación de las superficies mediante funciones matemáticas. El problema que presenta es que tiende a realizar un suavizado excesivo.

Estos datos se tendrán en cuenta a la hora de evaluar los errores obtenidos en las diferentes comparaciones entre nubes de puntos.

- 2D ½ TRIANGULATION (Triangulación de Delaunay)

Se presenta como otra de las alternativas en el ajuste y aproximación de las nubes de puntos. La triangulación de Delaunay es un método ampliamente utilizado en la generación de mallas tanto en 2D como en 3D.

Los métodos de ajuste basados en mallas triangulares estiman la normal en un vértice V_i a partir del promedio de las normales a cada una de las caras o triángulos adyacentes a V_i . La triangulación inicial es estimada mediante los métodos de Delaunay/Voronoi. Se muestra que los vértices más alejados proveen una buena estimación de la normal. Sin embargo, este método supone que no existe ruido en los datos.

La triangulación de Delaunay queda definida de la siguiente manera:

Sea X un conjunto de puntos del espacio euclídeo tridimensional E^3 , se define un poliedro de Voronoi $V(x_i)$, asociado al punto $x_i \in X$, como el conjunto de puntos del espacio E^3 que están más cercanos a x_i que a cualquier otro punto de X :

$$V(x_i) = \left\{ x \in E^3 \mid \frac{x}{d(x, x_i)} \leq d(x, x_j), \quad \forall x_j \in X, \quad j \neq i \right\}$$

Donde $d(x,y)$ es la distancia euclídea entre los puntos x e y . El poliedro de Voronoi $V(x_i)$ será un conjunto de puntos no acotado si y sólo si x_i se encuentra sobre la frontera de la envolvente convexa de X . El diagrama de Voronoi, por tanto, estará formado por el conjunto de los poliedros de Voronoi asociados a los puntos de X .

Existen subconjuntos R de cuatro o más puntos de X , no todos coplanarios, tal que un punto $v_R \in E^3$, equidistante a todos los puntos de R , satisface $d(v_R, x_k) < d(v_R, x_i)$, donde $x_k \in R$ y $\forall x_i \in X - R$.

La envolvente convexa de R recibe el nombre de politopo de Delaunay (poliedro convexo acotado) $D(R)$. Todos los puntos de R se encuentran dentro de una esfera, sin contener ninguno en el interior. Estarán formados normalmente por cuatro puntos, describiendo un tetraedro.

La triangulación de Delaunay se define como el conjunto de puntos X , $D(X)$, formado por los politopos de Delaunay.

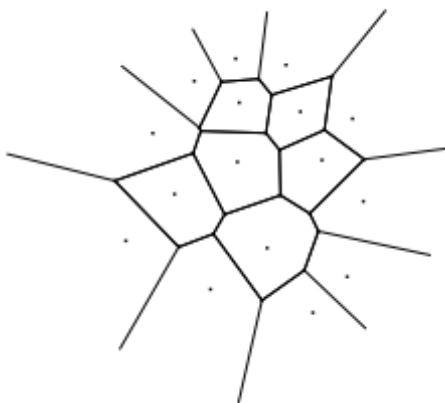


Figura 28. Diagrama Voronoi



Figura 29. Triangulación Delaunay

Una de las propiedades más interesantes que presenta la triangulación de Delaunay es que maximiza los ángulos interiores de los triángulos del modelo.

- HEIGHT FUNCTION (Función de altura cuadrática):

La función de altura cuadrática es simplemente un modelo cuadrático $Z = a + bx + cxy + dx^2 + ey + fy^2$. Es mucho más preciso que los planos, pero no maneja bien los bordes.

Construye un enmallado local basado en diagramas 3D de Voronoi. A través de una regla heurística propone un crecimiento de mallas para ajustar un grupo de curvas cuadráticas, mediante las cuales se obtienen los vectores tangentes.

La limitación que presenta este tipo de método es que no describe un manejo de la normal en bordes ni esquinas.

Representa la solución más adecuada en el caso de poseer una nube de referencia limpia, pero con alta curvatura. Este proceso matemático es el más preciso de todos los comentados hasta ahora, sin embargo, también es el que mayor tiempo de cálculo emplea.

Independientemente del método de comparación que se emplee se mostrarán en todos los casos tres gráficos:

- Información gráfica del error.
- Histograma de error.
- Valores de error gaussiano.

Gracias a estos gráficos y los resultados que proporcionan se podrán llevar adelante los análisis pertinentes.

Los valores referidos a la media y la desviación estándar gaussiana en la comparación entre nubes, para cada uno de los métodos, devuelven el módulo del vector de error. Este valor es absoluto, por lo que es necesario compararlo utilizando medidas robustas.

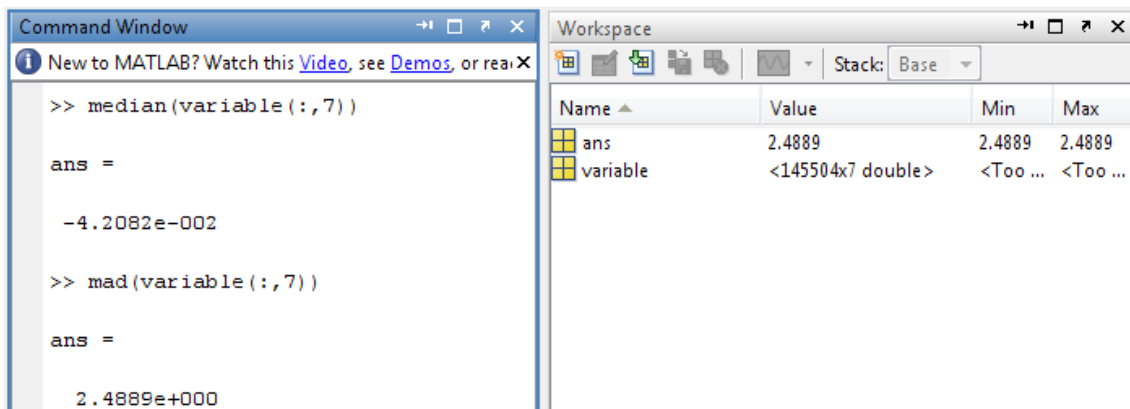
Para conseguir estas medidas robustas se emplea MATLAB. Mediante unas sencillas instrucciones, este software de programación nos facilita unas estimaciones robustas de la media y la desviación estándar.

El proceso consta de tres pasos:

1. Exportación de la nube que tiene el campo escalar en CloudCompare en formato .ASC.

2. Importación de la nube en Matlab.

3. La variable creada contendrá 7 columnas (x, y, z, r, g, b, campo escalar). El cálculo de las variables deseadas se hace mediante dos instrucciones: `median(variable(:,7))` y `mad(variable(:,7))`. La primera de ellas nos da la mediana, mientras que la segunda la diferencia absoluta de medianas. Ambas son estimaciones robustas de la media y desviación típica respectivamente.



```
>> median(variable(:,7))
ans =
-4.2082e-002

>> mad(variable(:,7))
ans =
2.4889e+000
```

| Name | Value | Min | Max |
|----------|-------------------|----------|----------|
| ans | 2.4889 | 2.4889 | 2.4889 |
| variable | <145504x7 double> | <Too ... | <Too ... |

Figura 30. Comando Matlab

En cuanto a las comparaciones entre las mallas, éstas se ejecutan a través de la herramienta **Compute Cloud/Mesh distance**. Los archivos con los que se trabaja son los modelados exportados en formato .STL.

CloudCompare no permite analizar malla frente a malla, sino que compara una nube de puntos con a una malla. Para cada punto de la nube busca el triángulo más cercano en la malla de referencia.

En nuestro caso, a la hora de llevar a cabo la comparación entre las dos mallas, el programa analiza la distancia desde los vértices de la malla del brazo con respecto a la superficie del modelado.

5. RESULTADOS

En este apartado se muestran los distintos resultados parciales y finales que se han ido obteniendo en las diversas fases, así como los de las comparaciones pertinentes entre las nubes y mallas.

Posteriormente, como se indica en el Anexo II se incluirán distintos archivos relativos a los resultados finales.

5.1 Nubes de puntos

Siguiendo el mismo guión que en este proyecto, se exponen en primer lugar las nubes de puntos de los dos equipos.

A continuación se mostrarán las notables diferencias que existen entre las nubes capturadas y posteriormente procesadas, del gaming sensor frente a las del brazo robótico.

- Nubes de puntos Kinect 2

El modelo generado a partir de los puntos obtenidos por la Kinect 2 engloba los 360º del motor. Por ello, resulta imprescindible mostrar las diferentes caras, ya que en función de la geometría que describe, se consiguen mejores o peores resultados.

En este apartado se incluyen las vistas: frontal, trasera, perfil izquierdo, perfil derecho y planta. Para un mayor detalle, se debe consultar el archivo adjunto “Nube_puntos_Kinect.asc”.



Figura 31. Vista frontal nube puntos Kinect



Figura 32. Vista trasera nube puntos Kinect



Figura 33. Perfil izquierdo nube puntos Kinect



Figura 34. Perfil derecho nube puntos Kinect

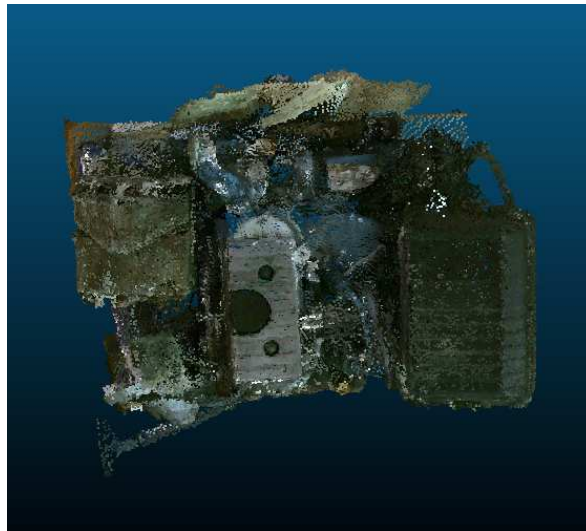


Figura 35. Planta nube puntos Kinect

La nube final, tras su tratamiento y optimización, consta de un total de 577.691 puntos.

Como se puede observar a simple vista, la nube obtenida posee bastante ruido, lo que se traduce en una pérdida de detalle. Esto es debido tanto al error ocasionado por la Kinect 2, como durante el procesado y alineamiento llevado a cabo por el propio usuario. Sin embargo, si tenemos en cuenta la gran cantidad de ruido que poseían las capturas iniciales, sin tratamiento alguno (Figura 10), se puede decir que se ha llevado a cabo un gran trabajo de depuración y limpieza.

Comparando las distintas vistas ofrecidas, resalta negativamente el perfil derecho, ya que en éste aparecen bastantes zonas de inexactitud. Los elementos pequeños, que se cruzan u ocultan parcialmente con otros, llevan a la confusión de los puntos adquiridos. Generan regiones con ausencia de claridad tanto para la limpieza del ruido como la posterior alineación.

Otra de las vistas que pierde cierto detalle en algunas zonas es la planta. Mientras que los elementos claramente definidos por su tamaño y geometría muestran unos resultados óptimos, la zona que coincide con la continuación del perfil derecho, en la cual se encuentran diversos componentes de pequeña envergadura (tubos), carece de esa exactitud.

- Nube de puntos brazo robótico

A diferencia de la nube de puntos de la Kinect, el trabajo desarrollado con el brazo robótico únicamente ha tenido lugar en una región del motor. Esta zona se corresponde con la cara que se ha denominado como frontal y parte de la superior.

Con el brazo robótico se consiguen unos resultados similares por todas las zonas en las que se han realizado pasadas. La calidad de los puntos obtenidos es análoga sobre la mayor parte de las regiones.

Otra de las características de estos datos es que el brazo robótico tan sólo captura las coordenadas de los puntos, no posee una cámara RGB como la Kinect para otorgar un color a cada uno de ellos. Por esta razón, la nube que se muestra aparece en un solo color, blanco en este caso.

El número total de la nube de puntos, tras su procesamiento, en el que se incluye la limpieza del ruido, alineamiento de las pasadas y optimización de los resultados obtenidos asciende a 1.014.817 puntos.

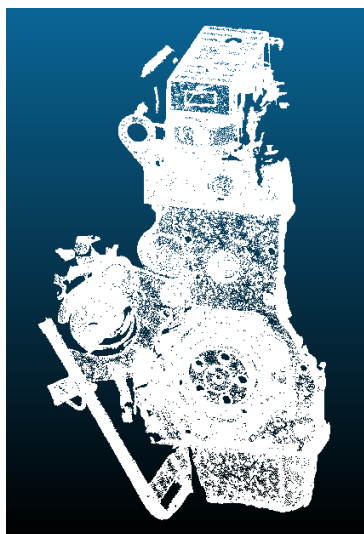


Figura 36. Nube de puntos brazo robótico

Si se compara proporcionalmente la cantidad de puntos alcanzados por uno y otro medio, frente al área que se ha estudiado con cada uno de ellos, destaca la gran cantidad de puntos generados por el brazo robótico respecto a los del gaming sensor. Pero la diferencia no sólo reside en la cantidad proporcional de puntos sino en la calidad de los mismos, y la definición que nos ofrecen del objeto analizado.

Esto se debe fundamentalmente a dos factores: error del equipo y error humano en el procesamiento. Mientras que el error descrito por el brazo robótico es prácticamente nulo en los niveles que se está trabajando, unos 0,0084 mm de precisión en el escaneo, el error propio de la Kinect podría alcanzar los 3 mm (Gráfico 1).

Por esta razón, a la hora de hablar y comparar los errores generados, tomaremos como modelo de referencia verdadera, o que al menos más se aproxima a la realidad, la nube de puntos y la malla desarrollada por el brazo robótico.

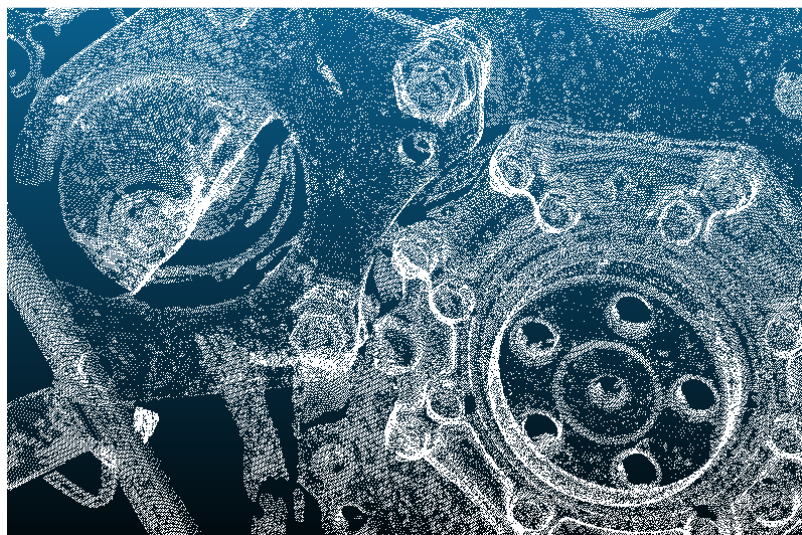


Figura 37. Detalle nube de puntos brazo robótico

Para una mayor profundización de los detalles que contiene la nube de puntos se incorpora el archivo “Nube_puntos_brazorobotico.asc”.

5.2 Mallas poligonales

A partir de las nubes de puntos anteriores se crean unas mallas poligonales, es decir, unos sólidos tridimensionales, siguiendo los pasos indicados en el apartado 3.2.1 y 3.2.2, relativos a los procesamientos de los correspondientes equipos.

- Modelo poligonal Kinect 2

La malla creada con RealWorks sirve de base para la generación del modelo sólido en 3D. Será el modelo poligonal en 3D el que se utilice para las comparaciones con las nubes de puntos y otras mallas.

En primer lugar se muestran los datos de los que se ha partido para la reconstrucción del modelo en 3D.

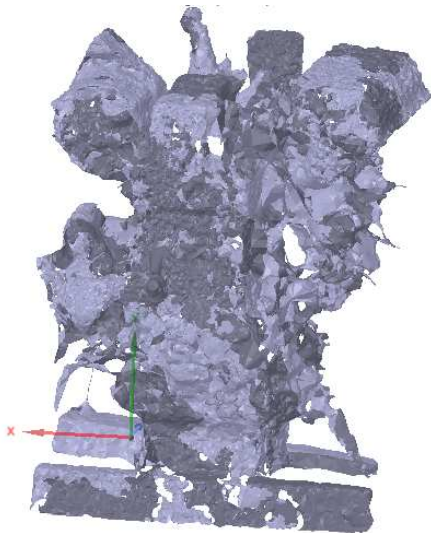


Figura 38. Vista frontal malla

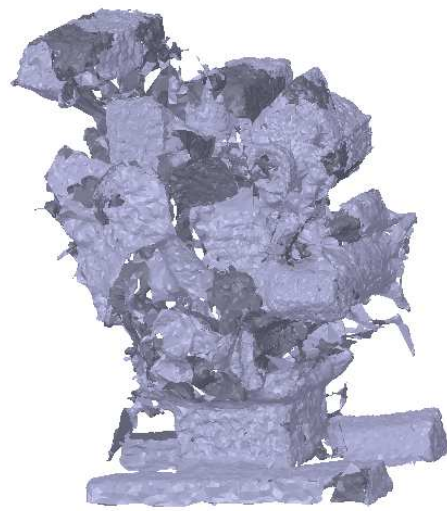


Figura 39. Vista trasera malla

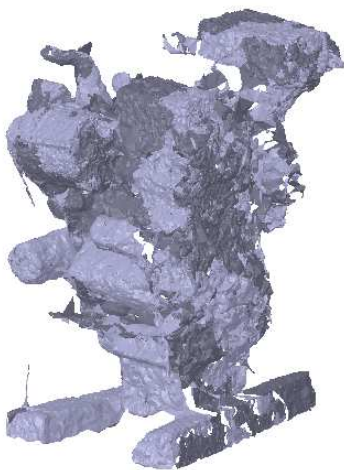


Figura 40. Vista lateral frontal malla

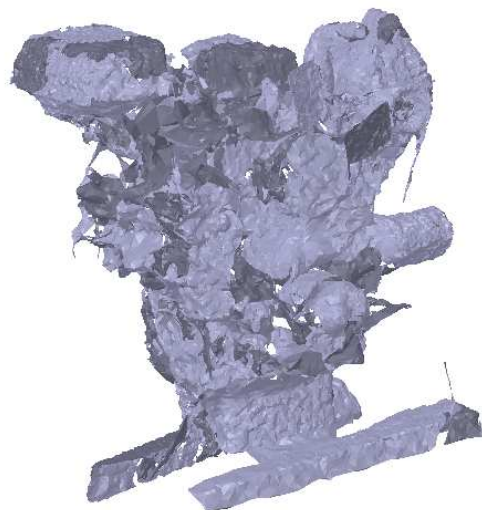


Figura 41. Vista lateral trasera malla

Como se puede comprobar, en algunas de las zonas hay ausencia de datos definidos, por lo que no es posible reconstruir plenamente el motor. En la generación del sólido, estas zonas quedarán vacías.

A partir de la malla anterior, y haciendo uso de las herramientas que proporciona el software de Geomagic se intenta reconstruir el sólido en 3D, manteniéndose lo más fiel posible a la realidad. Si bien es cierto, que se tiende a idealizar y por ello, el error acumulado hasta este momento, sigue en aumento.

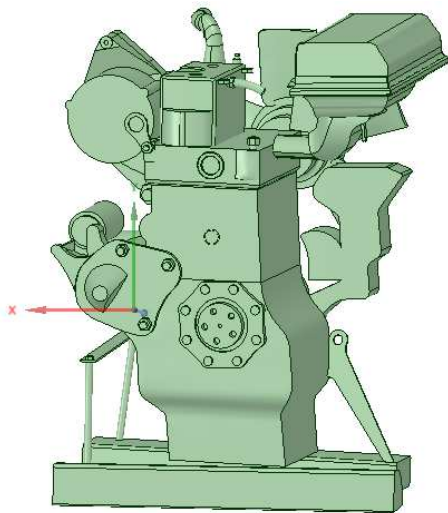


Figura 42. Vista frontal sólido 3D

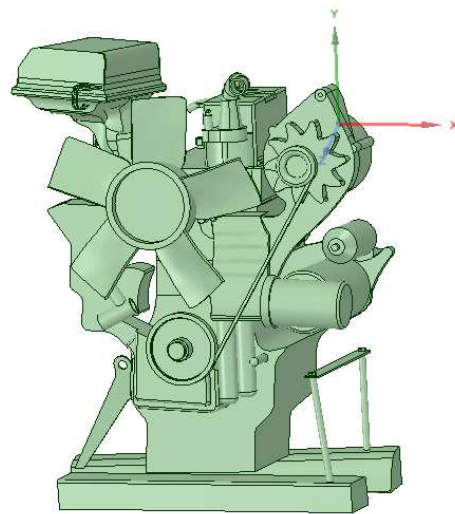


Figura 43. Vista trasera sólido 3D

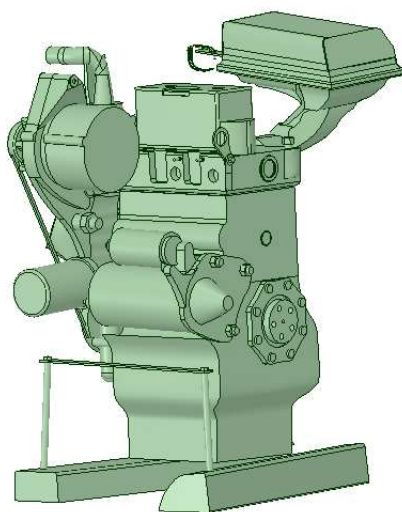


Figura 44. Vista lateral frontal sólido 3D

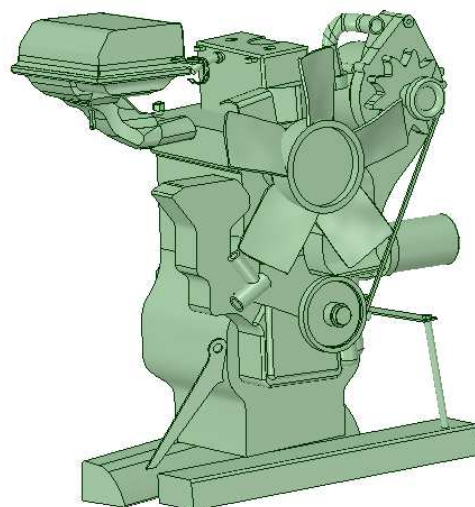
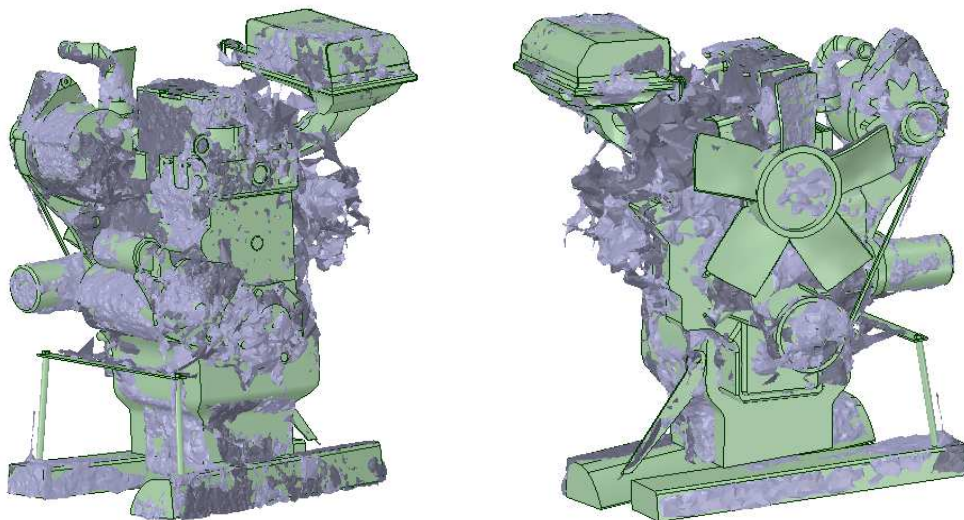


Figura 45. Vista lateral trasera sólido 3D

Al igual que ocurre con la nube de puntos, existen varias zonas en las que no se disponen de los datos suficientes. Por ello, en estas regiones se generará un vacío. El principal sector en el que tiene lugar este fenómeno se corresponde con uno de los laterales, concretamente el derecho. Del mismo modo, en numerosas zonas interiores se carece de la información necesaria.

A pesar de la escasa calidad de los datos de entrada, el modelo final presenta bastante similitud respecto al objeto real. Para comprobar los errores generados, tanto del equipo como los propagados durante el alineamiento, se comparará posteriormente este modelo con el poligonal obtenido a través del brazo robótico. En dicha comparación se pondrá de manifiesto las diferencias existentes entre un gaming sensor y un brazo robótico.

En las siguientes figuras se muestra la concordancia y disparidad entre la malla que se ha tomado como punto de partida, y el modelo sólido generado finalmente a partir de ella.



Figuras 46 y 47. Malla base y sólido 3D

Sin embargo, estos archivos no nos permiten hacer una comparación real de los errores. Para ello habrá que exportar el modelo en un formato .STL y poder así, en CloudCompare confrontarlo con las nubes de puntos y la malla poligonal generada por el brazo robótico.

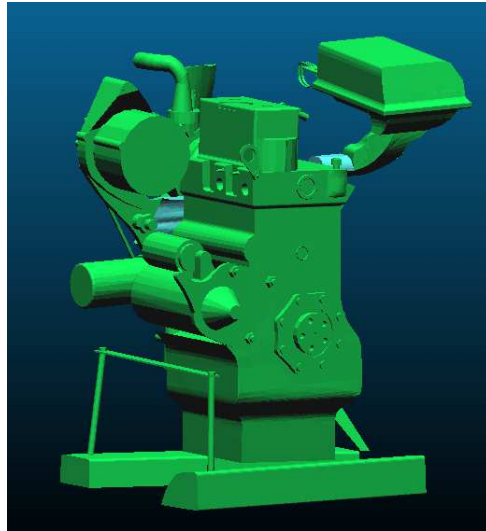


Figura 48. Modelo poligonal motor

Los archivos relativos al modelado del motor serán incluidos en el Anexo II con los nombres de: Modelo_poligonal_Kinect.stl.

- Modelo poligonal brazo robótico

Observando la cara frontal y la vista en planta del escaneo se que ha llevado a cabo con el brazo robótico en las siguientes imágenes, se puede percibir a simple vista que los resultados presentan una calidad muy superior a los arrojados por el gaming sensor.



Figura 49. Vista frontal modelo brazo robótico

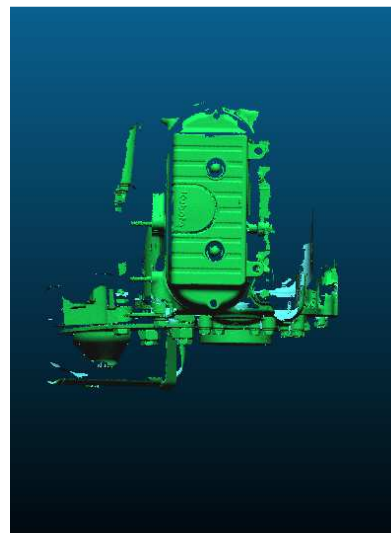


Figura 50. Vista en planta modelo brazo robótico

Si analizamos más profundamente se puede contemplar hasta las letras de la marca del motor (Kubota) grabadas en la parte superior de una de las piezas. Del mismo modo se pueden observar cada una de las tuercas que forman parte de la pieza octogonal o las oquedades existentes en la cara frontal.

Aquí es donde se contemplan las grandes disimilitudes que se producen en los resultados. Este nivel de detalle es totalmente imposible conseguirlo con la Kinect 2. La calidad ofrecida por un equipo y otro queda claramente en evidencia.

Se pondrá magnitud a los errores y diferencias registradas más adelante con la comparación de los distintos productos.

El modelo poligonal es también un .STL que será facilitado en el Anexo II a través del archivo Modelo_poligonal_brazorobotico.stl.

5.3 Comparaciones

Las comparaciones que se llevarán a cabo, como se ha indicado en apartados anteriores se realizarán por diferentes métodos en cada uno de los casos propuestos.

Antes de nada hay que importar los distintos archivos con los que se va a trabajar en formatos compatibles con CloudCompare y en las mismas unidades.

Para evitar una maximización de los errores, tras alinear los distintos archivos, serán recortados, de tal modo que sean coincidentes en su plenitud.

5.3.1 Nube de puntos individual (Kinect) vs. Nube de puntos global (Kinect) (NI(K)-NG(K))

La nube individual escogida es la que se ha tomado de referencia en el comienzo de las alineaciones. Por tanto, no será preciso realizar un alineamiento entre las nubes ya que son coincidentes.

Al ser esta nube individual el inicio de la nube global, se ha tomado como la nube de referencia.

En este caso, los errores de distancia mostrados, no pueden deberse al fallo de la Kinect 2 durante su captura, ya que ambas nubes han sido obtenidas a partir del mismo medio. Por tanto, la inexactitud acaecida es producto de la propagación de errores durante el alineamiento.

| Approx. results | | |
|-----------------|--------------------|-------------------------------------|
| 1 | Min dist. | 0 |
| 2 | Max dist. | 0.0358331 |
| 3 | Mean dist. | 0.00193383 |
| 4 | Sigma | 0.00310107 |
| 5 | Max relative error | 3.77496 + 0.413459/d % (d>0.0041... |

Tabla 2. Resultados aproximados (NI(K)-NG(K))

Una nota importante es que en este apartado todas las unidades vendrán dadas en metros. Sin embargo, en futuras comparaciones, las medidas serán directamente en milímetros.

Analizando los resultados, se deduce que el error medio es de unos 2 mm. No hay que prestar la mayor importancia al dato de distancia máxima, ya que posiblemente haya sido causada por la introducción de unos puntos extra que no están en ambas nubes.

Además de los resultados globales anteriores, se analizan los obtenidos por cada uno de los métodos. Para evaluar correctamente los errores generados en cada uno de los procedimientos, se compararán los valores de media y desviación estándar gaussiana con los obtenidos a través de las estimaciones robustas de las mismas del campo escalar. Estas estimaciones se llevan a cabo utilizando Matlab, siguiendo el proceso descrito en el apartado “4. COMPARACIÓN EQUIPOS”.

Las medidas robustas de la media y la desviación típica gaussianas que nos ofrece el campo escalar, obtenidas a través de Matlab (mediana y mad) son las siguientes:

| | (m) |
|---|---------|
| Mediana | 0,00138 |
| Mad (Diferencia absoluta medianas) | 0,00190 |

Tabla 3. Estimaciones robustas

Entre los gráficos resultantes tras la aplicación de los diversos métodos se encuentran los que se muestran a continuación:

- Información gráfica del error.
- Histograma de error.
- Valores de error gaussianos.

- *None:*

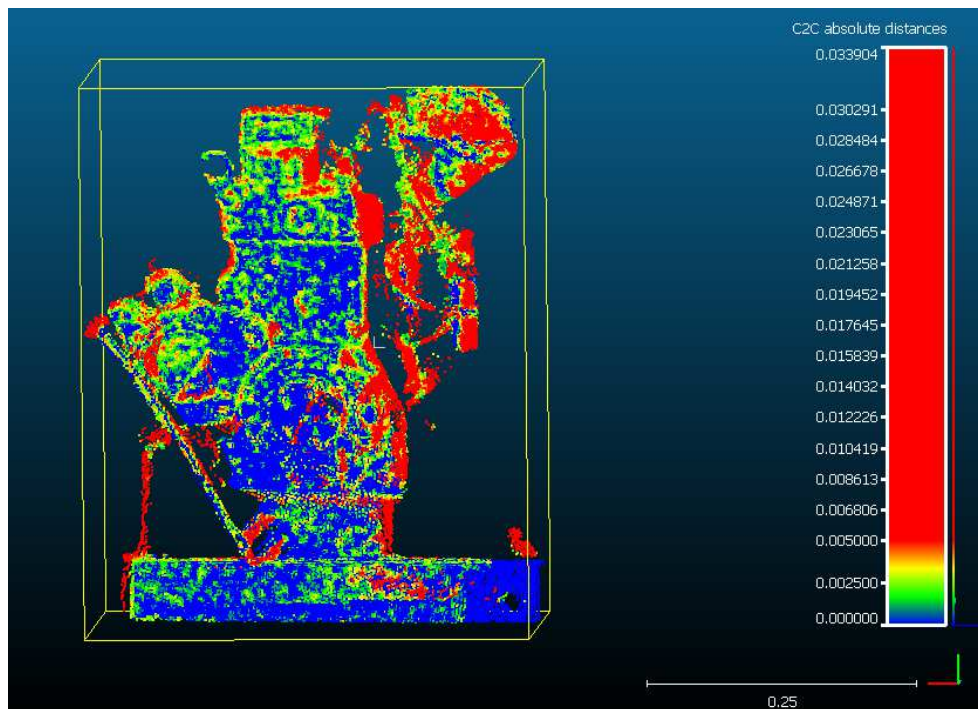


Figura 51. Información gráfica error (None)

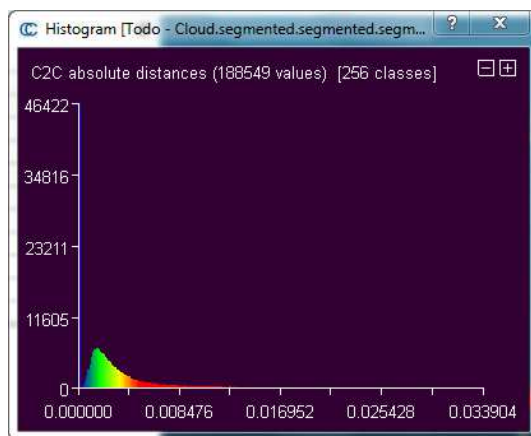


Figura 52. Histograma de error (None)

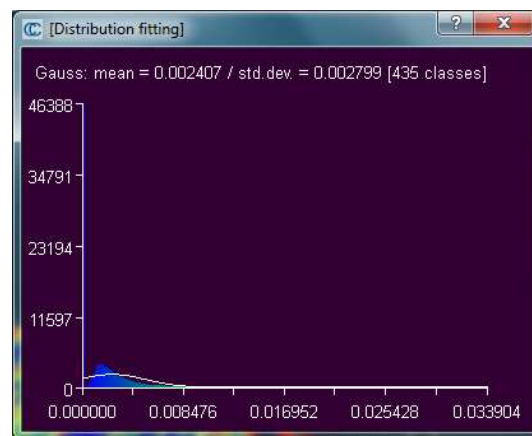


Figura 53. Valores de error gaussianos (None)

| | NONE | (m) |
|---------------------|------|---------|
| Media | | 0,00241 |
| Desviación estándar | | 0,00280 |

Tabla 4. Valores gaussianos NONE

- *Least Square Plane:*

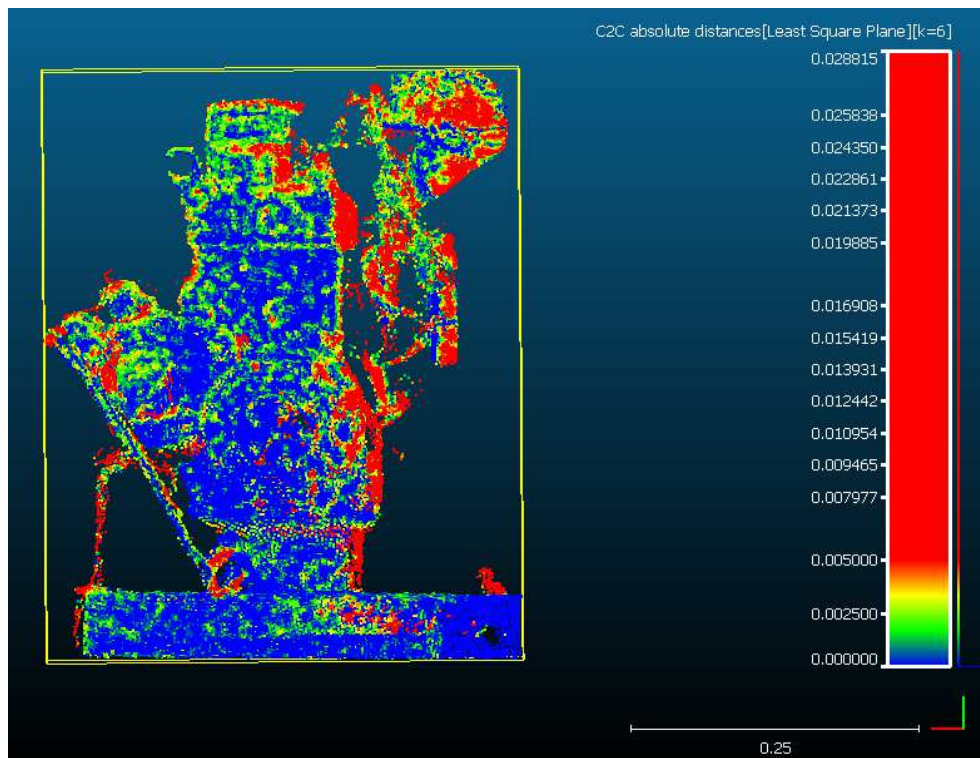


Figura 54. Información gráfica error (Least Square Plane)

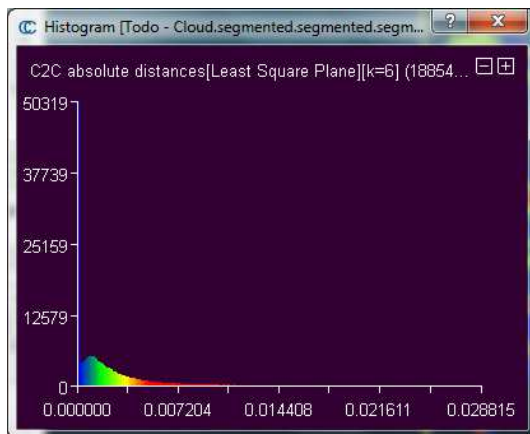


Figura 55. Histograma de error (Least Square Plane)

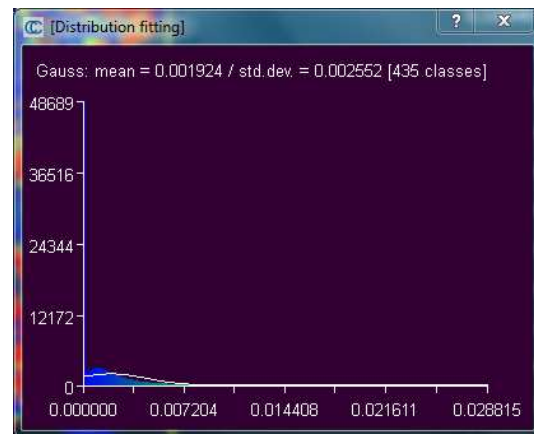


Figura 56. Valores de error gaussianos (Least Square Plane)

| LEAST SQUARE PLANE | (m) |
|---------------------|---------|
| Media | 0,00192 |
| Desviación estándar | 0,00255 |

Tabla 5. Valores gaussianos LEAST SQUARE PLANE

- 2D ½ Triangulation

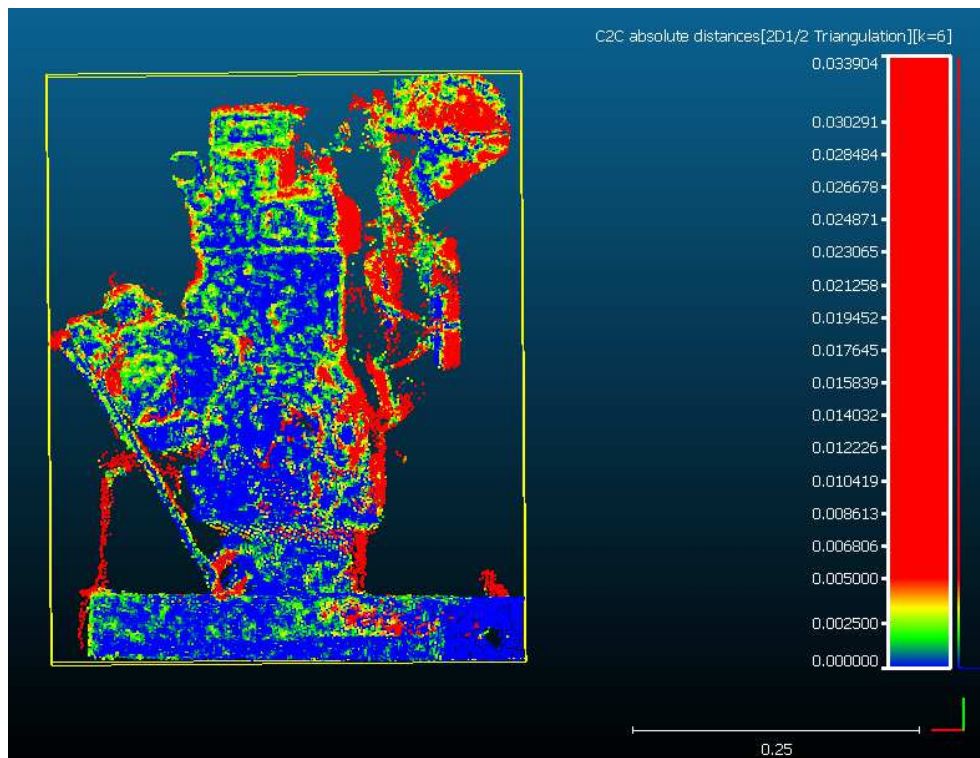


Figura 57. Información gráfica error (2D ½ Triangulation)

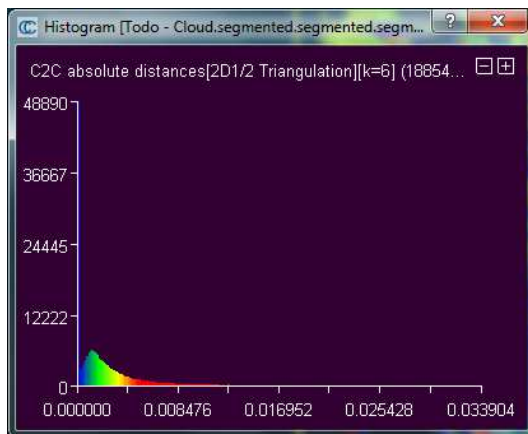


Figura 58. Histograma error (2D ½ Triangulation)

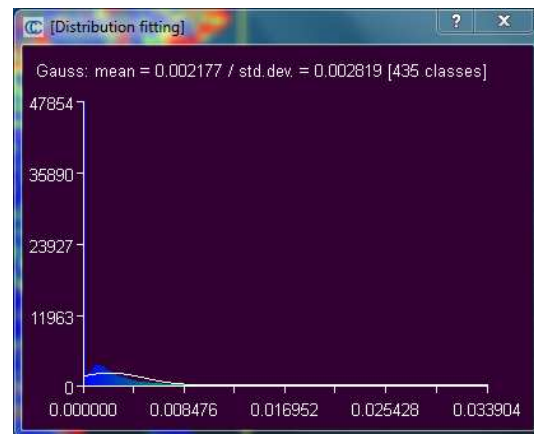


Figura 59. Valores de error gaussianos (2D ½ Triangulation)

| 2D ½ TRIANGULATION | (m) |
|---------------------|---------|
| Media | 0,00218 |
| Desviación estándar | 0,00282 |

Tabla 6. Valores gaussianos 2D ½ TRIANGULATION

- *Height Function:*

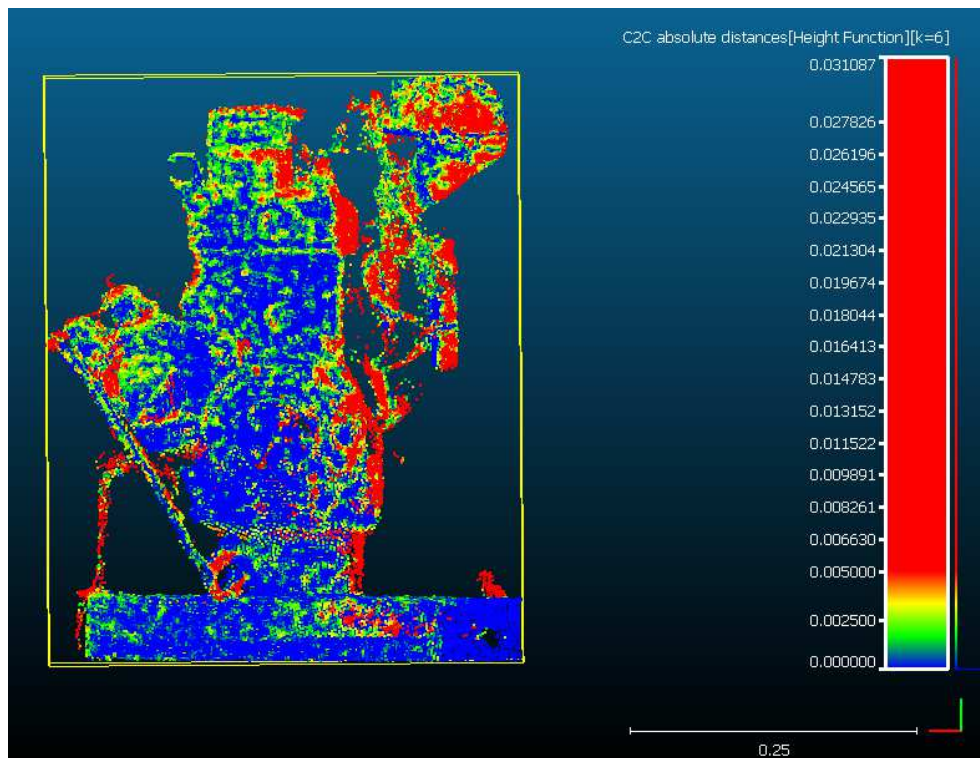


Figura 60. Información gráfica error (Height Function)

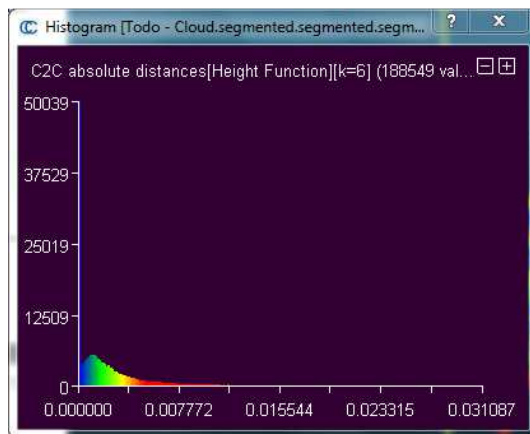


Figura 61. Histograma error (Height Function)

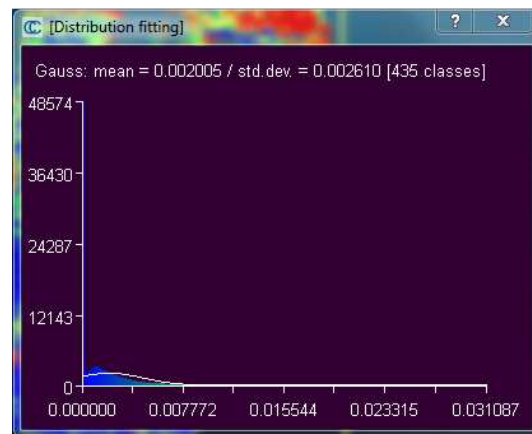


Figura 62. Valores de error gaussianos (Height Function)

| HEIGHT FUNCTION | (m) |
|---------------------|---------|
| Media | 0,00201 |
| Desviación estándar | 0,00261 |

Tabla 7. Valores gaussianos HEIGHT FUNCTION

Las gráficas con los valores gaussianos no se corresponden con una distribución gaussiana corriente. Por esta razón, se hace necesario obtener las estimaciones robustas, y poder así comparar los errores.

En la información gráfica del error aparecen en color rojo aquellas zonas que superan una distancia de 5 mm. Como se puede comprobar, en función del método que se haya empleado existirán mayores o menores zonas con este tono.

Analizando los resultados se puede observar que el método None (búsqueda del vecino más cercano) es el que mayores distancias presenta entre las nubes de puntos. Queda reflejado tanto en la gráfico donde se ve una mayor tonalidad amarillenta o verdosa y, por supuesto, en los valores del error gaussianos. Las diferencias de estos resultados con respecto a las estimaciones robustas son las más elevadas, pasando de los 0,001378 metros a los 0,00241 metros en la media por ejemplo. Esto es debido a la presencia de ruido en las nubes.

Respecto al resto de procedimientos desarrollados, el que más se aproxima a los valores robustos es Least Square Plane (Ajuste por mínimos cuadrados). Este método resulta el más adecuado en el caso de poseer nubes con ruido y valores atípicos, como es el nuestro.

Independientemente del método, se observa en todos los gráficos que tanto el cuerpo central como el apoyo de madera, son las zonas en las que más se aproximan las nubes de puntos. Sin embargo, el elemento localizado en la parte superior así como un saliente situado en la izquierda son las regiones donde mayores disparidades se encuentran. Una de las razones puede ser el material de dichas piezas, que es distinto al del resto.

En estas comparaciones no se están teniendo en cuenta las zonas rojas, que se corresponden con bordes, esquinas o las piezas de la derecha, en las que existe una gran variabilidad en la cantidad de puntos recogidos en ambas nubes. La mayor parte de ellas están localizadas en la parte derecha de la imagen. La causa de ello es la perspectiva desde la cual se tomó la captura de la nube individual, quedando ciertas regiones de sombra o ausencia de información. Este problema no ocurre en la nube global porque ha sido completada por el resto de nubes y no tiene, por tanto, un único punto de vista.

El archivo con los mapas de color se incluye en el Anexo II: "Nube individual vs Nube global.bin".

5.3.2 Nube de puntos (Kinect) vs. Malla modelado 3D

Inicialmente la nube y la malla se encuentran alineadas, puesto que la segunda ha sido obtenida a partir de la primera. Por tanto, los errores generados son debidos a la idealización que se realiza durante la reconstrucción tridimensional.

En esta fase se determina la distancia de cada punto de la nube a la superficie del modelo. Debe tenerse en cuenta que las unidades en las que se trabaja en este apartado y en los siguientes son milímetros.

| Approx. results | | |
|-----------------|--------------------|--|
| 1 | Min dist. | 0 |
| 2 | Max dist. | 137.153 |
| 3 | Mean dist. | 6.55214 |
| 4 | Sigma | 14.2584 |
| 5 | Max relative error | $3.77496 + 447.237/d \%$ ($d > 4.47237$) |

Tabla 8. Resultados aproximados (N(K)-Modelo)

| | (mm) |
|------------------------------------|--------|
| Mediana | 1,9789 |
| Mad (Diferencia absoluta medianas) | 7,8412 |

Tabla 9. Estimaciones robustas (Nube-Modelado)

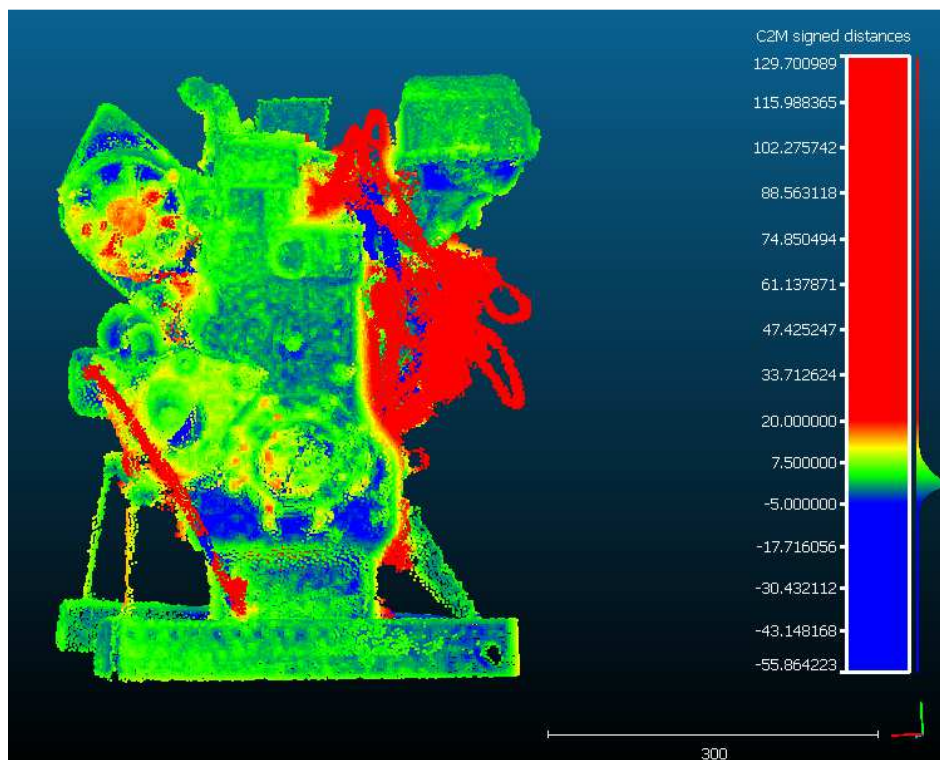


Figura 63. Información gráfica del error Nube vs. Modelado

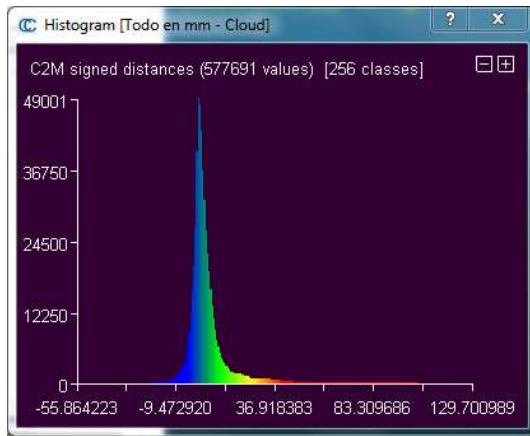


Figura 64. Histograma error (Nube-Modelado)

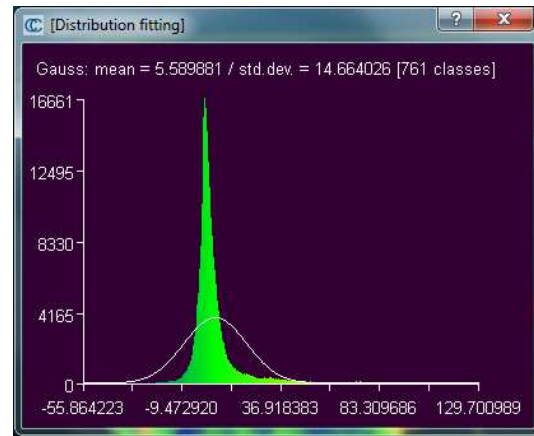


Figura 65. Valores de error gaussianos (Nube-Modelado)

El rango de saturación de la información gráfica del error nos permite jugar con la visualización. Las distintas tonalidades reflejadas nos posibilitan la estructuración de las regiones en tres grupos distintos:

- Rojo: En este color se incluyen los puntos que se encuentran a una distancia mayor de 2 cm. Al tratarse de una comparación entre la nube de puntos y el modelado, nos indica que estas zonas no han sido reconstruidas en la recreación tridimensional.
- Amarillo: estas zonas sí que han sido reconstruidas pero existen disparidades entre la nube y el posterior modelado. Las diferencias se encuentran alrededor de 1 cm.
- Verde-azul: son regiones coincidentes en ambos objetos.

La imagen incluida en esta memoria sólo está referida a la cara frontal, el resto del modelado se incluye en un archivo adjunto: "Nube vs modelado.bin".

A diferencia de la comparación entre puntos, los gráficos de errores sí que muestran una distribución gaussiana, en los que la distancia media se encuentra a 5,58 mm, con una desviación estándar de 14,66 mm. Sin embargo, estos valores sufren ciertas variaciones si se comparan con los obtenidos en las estimaciones robustas a través de Matlab. Esto es debido a la asimetría que presenta la distribución. En el caso de la desviación estándar, debería aproximarse a 1,4826 veces Mad (Diferencia absoluta medianas) si fuese gaussiana. Sin embargo, este valor se aproxima a 11,62 mm frente a los 14,66 mm expresados en el gráfico.

Respecto a los errores hay que comentar que el error supuesto a priori de la nube de puntos de la Kinect es de $\pm 4,6$ mm, sin embargo el error final obtenido es

superior, $2 \text{ mm} \pm 7,8 \text{ mm}$. La explicación de ello es que no sólo está presente el error instrumental del sensor (3 mm), sino que hay que añadir los errores derivados de las etapas de generación de mallas y su posterior diezmado ($<3,5 \text{ mm}$, pág.24), así como el proceso de idealización de los componentes en primitivas geométricas (planos, cilindros,...). Además, hay que recordar que se está modelando una pieza real sujeta a deformación, por lo que el proceso de idealizar elementos contribuye al error.

Las mayores distancias, representadas en rojo, coinciden con la zona derecha en la que se encuentran los pequeños elementos que no han sido reconstruidos, comentados en otros apartados de esta memoria, llegando a estar hasta unos 13 cm de separación con el resto.

A raíz de este análisis se manifiesta que el error final de modelar con la Kinect 2 es centimétrico, es decir, del orden del sensor usado y, por tanto, coherente con la realidad. Todo esto en términos de un error de ajuste, concebido como el conjunto del error propio del sensor más el generado durante el procesado.

5.3.3 Malla brazo robótico vs. Malla modelado 3D

En este caso las dos mallas no se encuentran en el mismo sistema de referencia, por lo que será necesario realizar una alineación previamente a la comparación.

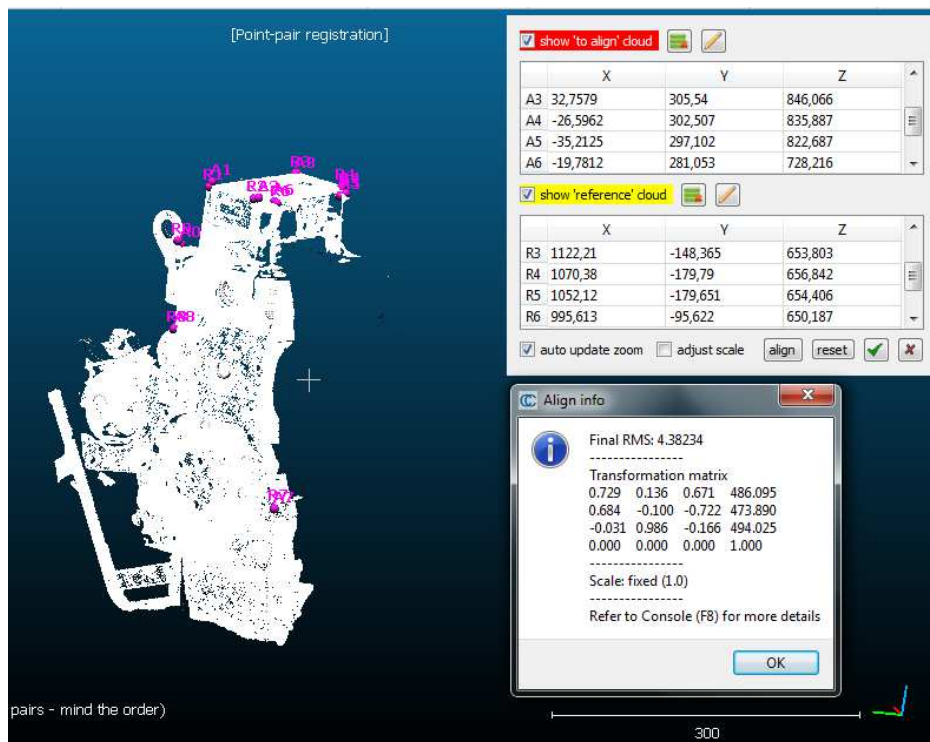


Figura 66. Alineación mallas

En el momento de alinear las dos mallas se provoca un aumento del error que venimos arrastrando. Este error de alineación, que en este caso es de 4,38 mm, deberá tenerse en cuenta en la posterior interpretación de los errores.

Una vez alineadas, se procede a la comparación de las dos mallas obteniendo los siguientes resultados.

| Approx. results | | |
|-----------------|--------------------|---|
| 1 | Min dist. | 0 |
| 2 | Max dist. | 26.1171 |
| 3 | Mean dist. | 3.11638 |
| 4 | Sigma | 3.75907 |
| 5 | Max relative error | $3.77496 + 340.657/d$ % ($d > 3.40657$) |

Tabla 10. Resultados aproximados (Malla brazo-Modelo)

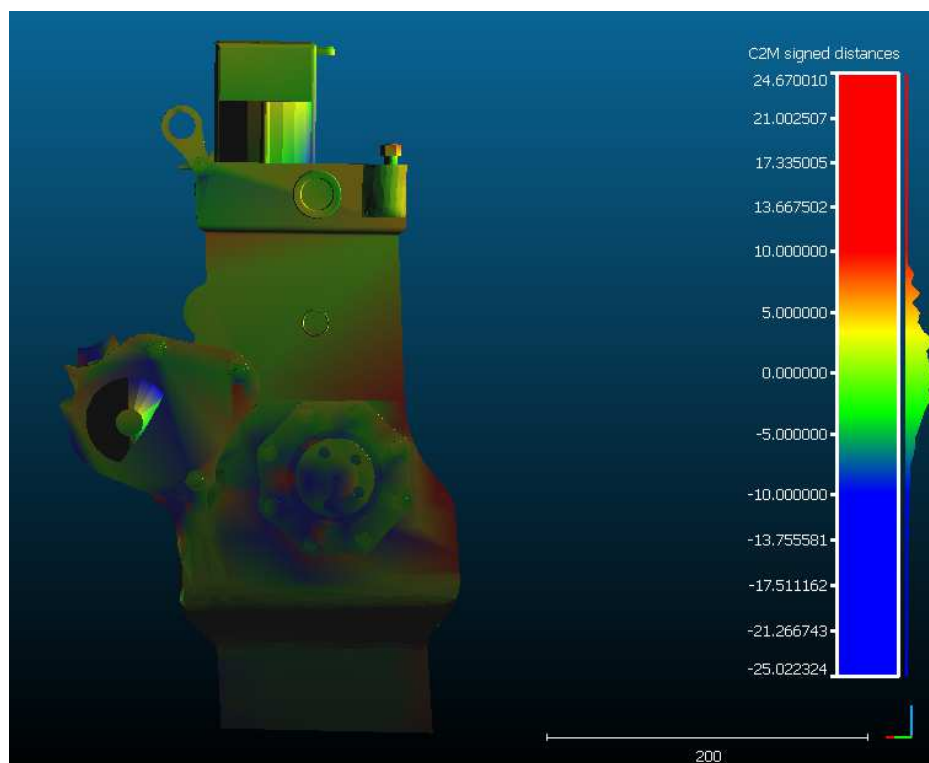


Figura 67. Información gráfica del error Malla brazo vs. Modelado

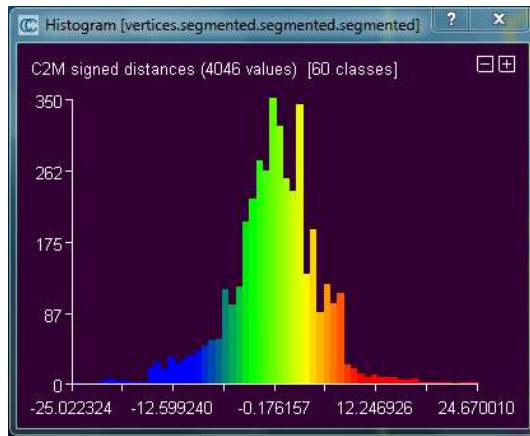


Figura 68. Histograma error (Malla brazo-Modelado)

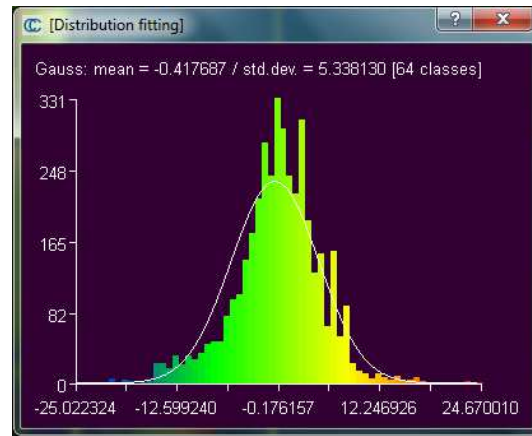


Figura 69. Valores de error gaussianos (Malla brazo-Modelado)

Las regiones de mayor disconformidad se corresponden con los tonos azules y rojos. Estos indican que en estas zonas, la distancia entre las mallas es superior a 1 cm, por lo que se puede decir que no hay concordancia.

Los sectores que más se alejan entre las dos mallas pertenecen a los dos elementos que sobresalen en la parte central de la cara frontal. Al tratarse de unas piezas con una geometría más complicada, el error que se ha venido acumulando desde el principio es mayor.

Si tenemos en cuenta, los errores que venimos arrastrando, debemos sumar por un lado el error esperado de la propia Kinect 2, el cual se estima en 3 mm. Dado el escaso error que genera el brazo robótico durante su captura y posterior procesado ($\pm 0,084$ mm), se puede asumir que éste es nulo.

Por otro lado, existirá el error acumulado durante las alineaciones de las distintas nubes de puntos, porque no hay que olvidar que el sólido con el que estamos trabajando parte de la nube de puntos global obtenida por la Kinect 2. Además se debe añadir el error generado por la idealización de los elementos a la hora de reconstruir el objeto.

Finalmente, y en último lugar, se adiciona la falta de precisión durante el alineamiento de las mallas que se van a comparar que se estima en 4,38 mm.

Por tanto, el error estimado a priori sería de 9,0 mm (composición cuadrática del error de ajuste del modelado [7,78 mm, que incluye el error del sensor y de procesado] y el error de referenciación del sistema coordenado [4,38 mm]).

El gráfico gaussiano nos indica que la distancia media se encuentra en -0,42 mm, con una desviación estándar de 5,34 mm. Si lo comparamos con los errores comentados anteriormente con los obtenidos finalmente, comprobamos que están

dentro del baremo esperado, incluso es inferior. Esto es debido a que tan sólo se está analizando una parte del modelo.

Por tanto, si se dispone de una verdad terreno, considerando a efectos prácticos que el error es nulo, la exactitud real del modelo es inferior al centímetro en la zona evaluada, considerando que la exactitud es la composición del error del ajuste real del modelo más el error de referenciación. Finalmente se obtiene un error de ajuste del modelo de unos 7,8 mm (precisión del modelo), pero en la zona donde se disponen datos reales, se alcanza una exactitud del modelo de 5,5 mm.

El archivo de comparación se encuentra incluido en el Anexo II: “Malla brazo vs Modelado 3D.bin”

5.3.4 Nube de puntos (Kinect) vs. Malla brazo robótico

Al igual que en el caso anterior, es necesario llevar a cabo una alineación entre los dos archivos, ya que proceden de dos medios distintos, la nube de la Kinect y la malla del brazo robótico.

Se tomará como referencia la malla del brazo robótico, al considerar su error nulo.

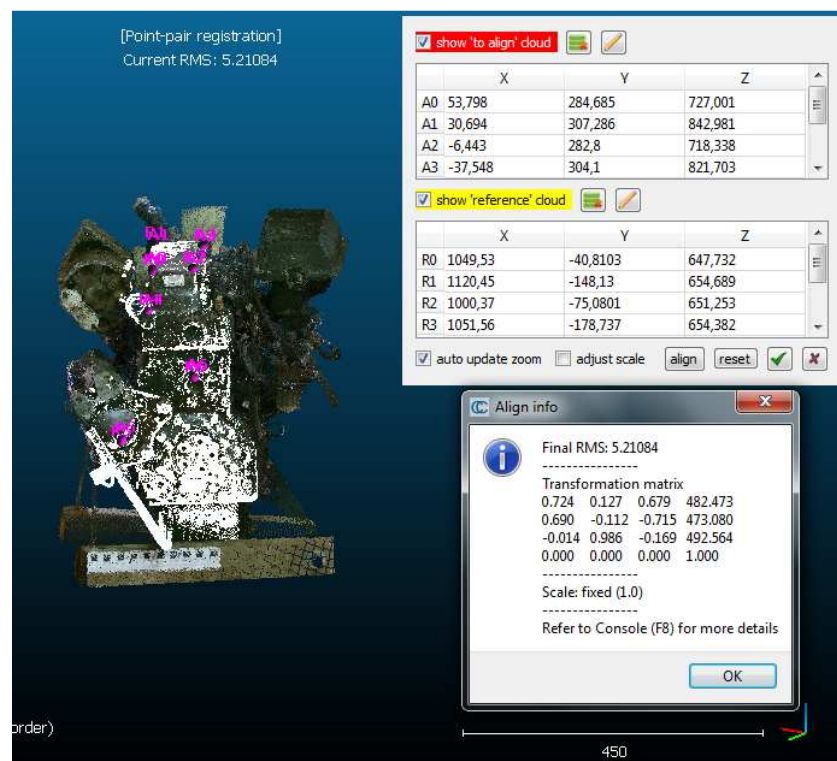


Figura 70. Alineación nube y malla

La alineación entre la nube de puntos de la Kinect y la malla poligonal del brazo robótico supone un aumento del error. Sin embargo, la suma de los errores en este caso será inferior que la del apartado anterior, ya que en este sólo intervienen el error propio del gaming sensor y la inexactitud provocada por las diversas alineaciones.

Esta será la mejor prueba de comparación entre los dos medios, aunque vaya implícito en las anteriores. Se podría decir que se equiparan directamente los datos obtenidos por cada uno de ellos, después de ser tratados y optimizados.

A continuación se muestran los resultados y gráficos obtenidos:

| Approx. results | | |
|-----------------|--------------------|--|
| 1 | Min dist. | 0 |
| 2 | Max dist. | 30.5098 |
| 3 | Mean dist. | 1.78676 |
| 4 | Sigma | 2.62305 |
| 5 | Max relative error | $3.77496 + 352.036/d \%$ ($d > 3.52036$) |

Tabla 11. Resultados aproximados (Nube-Malla brazo)

| | (mm) |
|------------------------------------|---------|
| Mediana | -0,0421 |
| Mad (Diferencia absoluta medianas) | 2,4889 |

Tabla 12. Estimaciones robustas (Nube-Malla brazo)

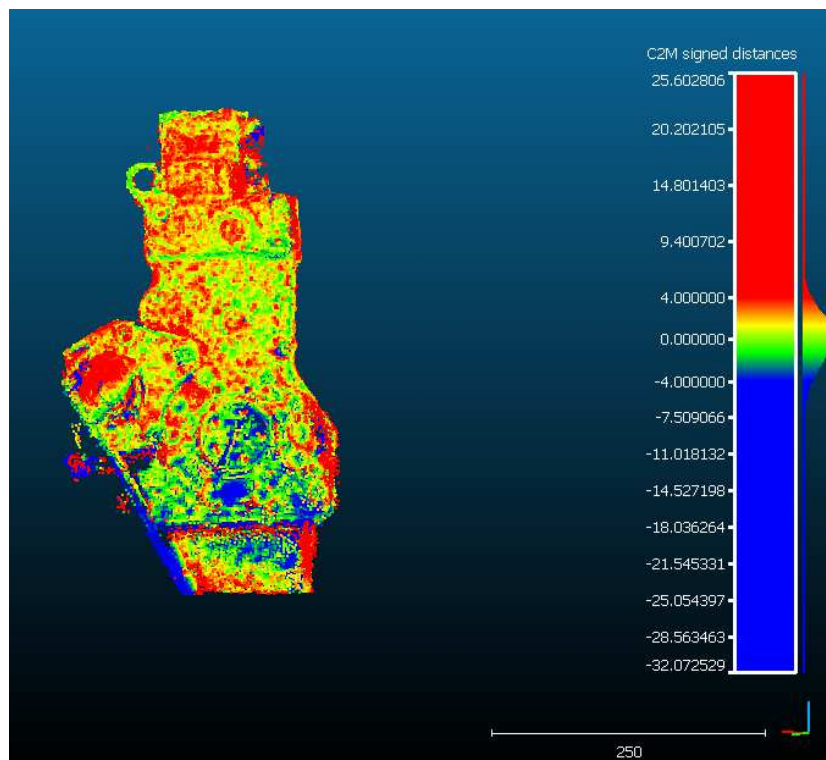


Figura 71. Información gráfica del error Nube Kinect vs. Malla brazo

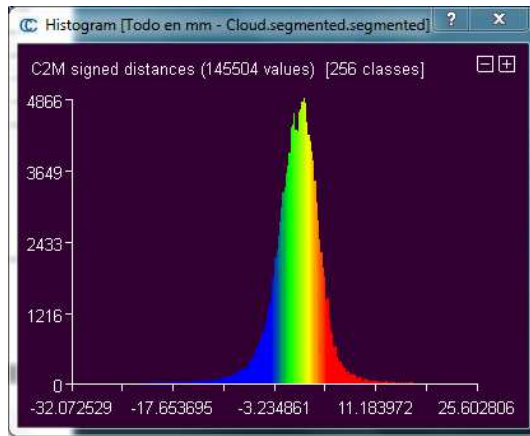


Figura 72. Histograma error (Nube Kinect-Malla brazo)

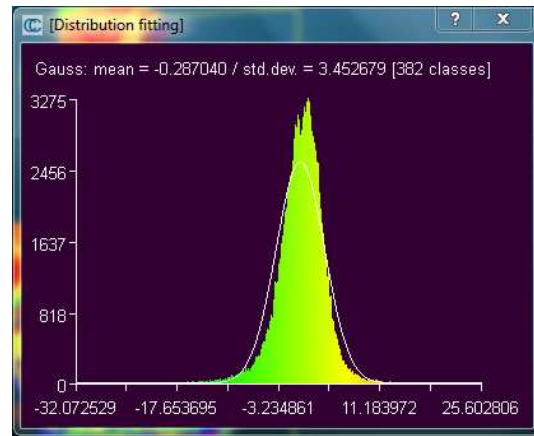


Figura 73. Valores de error gaussianos (Nube Kinect-Malla brazo)

De todas las comparaciones realizadas ésta es la que presenta un índice de error más bajo. Si observamos la información gráfica del error, la mayor parte de los puntos presentan una tonalidad amarillenta y verdosa, lo que nos indica que se encuentran a una distancia inferior a los 2 mm.

Dado que la parte de arriba posee tonos rojizos (valores positivos) y la de abajo un color más azulado (valores negativos), se puede interpretar que existe un error en el alineamiento entre la nube y la malla.

La curva descrita en el gráfico se ajusta notablemente a una gaussiana, de ahí que su media sea de $-0,29$ mm, con una desviación estándar de $3,45$ mm. Si comparamos estos datos con las estimaciones robustas, son bastante parejos: una media de $-0,04$ mm y una desviación estándar de $2,49$ mm. En ambos casos, valores inferiores a los resultados aproximados.

Todos los valores registrados nos indican que nos encontramos dentro del rango esperado de fallo, y responden tanto a la inexactitud propia del equipo como al desacierto del usuario en las alineaciones.

Finalmente, el archivo con los mapas de color será incluido en el Anexo II: "Nube vs malla brazo.bin".

6. CONCLUSIÓN

El objetivo que se planteaba al comienzo de este proyecto era el análisis de dos equipos completamente distintos. Se establecía como propósito valorar hasta qué punto resulta conveniente la utilización de uno u otro. Para ello se tendrán en cuenta diversos aspectos.

Por un lado existe el tema económico. Mientras que la Kinect está al alcance de cualquier persona, dado su bajo coste, el brazo robótico hay que enfocarlo en otra línea. Hay que estar seguro de que la inversión realizada en su compra resultará rentable. El propósito y la finalidad de un brazo robótico nada tienen que ver con el de un gaming sensor.

Otro de los aspectos a tener en cuenta es el tiempo invertido en el modelado. Las técnicas de modelado en ingeniería inversa, si bien son semiautomáticas y tienen asistentes para diferentes primitivas geométricas básicas, en el caso de modelizar elementos con formas libres o procedentes de sensores de baja precisión, el operador no puede ser reemplazado, según el estado de la técnica actual. Para el caso de estudio ejemplificado en este proyecto, la etapa de modelado fueron alrededor de 150 horas, en contraste con las 3 horas necesarias para la captura.

Más allá de los temas económico y temporal, se ha de valorar el resultado que estamos buscando a nivel de calidad. Las comparaciones realizadas nos permiten conocer la precisión final del modelo CAD obtenido a través de la Kinect, puesto que para efectos prácticos se ha considerado que el error generado por el brazo robótico es nulo.

A raíz de los resultados cosechados se puede concluir que el modelo CAD final tiene una precisión centimétrica o incluso mejor, dependiendo de la tipología de la zona. De modo secundario, se ha evaluado la exactitud del sensor Kinect 2 de forma general (fuera del contexto de la ingeniería inversa), obteniéndose valores inferiores a los 5 mm para una distancia de toma de 1m. Esto es significativo, tanto por la ausencia de datos sobre la calidad medida del sensor, debido a su novedad, como por la significativa mejora con respecto a la versión anterior, cuyo error era centimétrico.

Todo ello habilita al gaming sensor como un sensor adecuado en aplicaciones de ingeniería inversa que requieren modelizaciones a gran escala de forma rápida y bajo coste. Por ejemplo, la contextualización de detalle de los componentes de un aeroplano (fuselaje, hidráulicos,..), maquinaria industrial (volquetes de minería,...) del cual no se disponen de los planos constructivos, ya sea por restricciones comerciales o antigüedad de construcción de equipamiento, entre otros.

Si bien es cierto, que en ningún momento se puede pretender lograr el mismo nivel de detalle que ofrece un equipo especializado, como es un brazo robótico. Los errores por debajo del milímetro generados por este equipo, lo capacitan para actividades de ingeniería inversa que requieren una mayor finura y minuciosidad en su trabajo. Tiene numerosos campos de aplicación: industrial (piezas, elementos o engranajes que forman parte de mecanismos,...), médico (prótesis), forense, arquitectónico,...

El uso de gaming sensors en ingeniería inversa presenta una gran proyección de cara al futuro. La utilización de estos sistemas ligeros, que pueden trabajar en el exterior, con un tiempo reducido de captura, y por supuesto, de bajo coste, lo convierten en una gran alternativa para las empresas. La "limitación" centimétrica para algunos tipos de estudio se irá reduciendo conforme avance el tiempo y se vayan desarrollando nuevos equipos.

A parte de las restricciones actuales que puede presentar a nivel de precisión en algunos casos, resulta de gran interés la mejora del software de tratamiento de datos, tanto a nivel de alineamiento como en la detección de cuerpos sólidos irregulares a partir de una malla.

7. BIBLIOGRAFÍA

Apuntes:

- González Aguilera D. *Apuntes asignatura Procesamiento de datos láser y radar*. Universidad de Salamanca.
- Rodríguez González P. *Apuntes asignatura Procesamiento de datos de sensores híbridos*. Universidad de Salamanca.

Documentos:

- Bels, P.J., McKay, N. 1992. *A method for registration of 3-d shapes*. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 239-256, Vol. 14, nº2.
- Delfín Vázquez J.J., Jiménez López E. *Ingeniería Inversa: Metodología y aplicaciones*. Foro de análisis de investigación, desarrollo y gestión tecnológica en ITESCA.
- Gómez Martínez L. *Proyecto técnico-económico de utilización de técnicas de ingeniería inversa y prototipado rápido aplicadas al rediseño y mejora de productos*. Universidad Pontificia de Comillas. Agosto 2012, Madrid.
- Leal, E., Leal, N., Sánchez, G. *Estimación de normales y reducción de datos atípicos en nubes de puntos tridimensionales*. Información Tecnológica, pp. 39-46, Vol. 25, nº2.
- Moreno Seco, F., *Clasificadores eficaces basados en algoritmos rápidos de búsqueda del vecino más cercano*. Universidad de Alicante. Febrero 2004.
- Rodríguez Barrera, E., *Modelización y simulación numérica de campos de viento mediante elementos finitos adaptativos en 3D*. Universidad de Las Palmas de Gran Canaria. Mayo 2004.
- Torres G. *Ingeniería inversa: una herramienta para la iniciación al diseño*. V Congreso Bolivariano de Ingeniería Mecánica. 17-20 Junio de 2008, Cúcuta, Colombia.

Sitios Web:

- Brazo robótico:
 - www.hexagonmetrology.com [Fecha de consulta: Agosto 2014]
 - <http://www.pedeca.es/2014/05/05/nueva-generacion-de-romer-absolute-arm/> [Fecha de consulta: Agosto 2014]
 - <http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Hardware/scanner3D/Escaner3D.html> [Fecha de consulta: Agosto 2014]
- Kinect:
 - <http://blogs.msdn.com/b/esmsdn/archive/2011/07/20/reto-kinect-usar-las-c-225-maras-del-sensor.aspx> [Fecha de consulta: Agosto 2014]
 - <http://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx> [Fecha de consulta: Agosto 2014]
 - http://www.microsoft.com/en-us/kinectforwindows/purchase/sensor_setup.aspx [Fecha de consulta: Agosto 2014]
 - http://3dom.fbk.eu/files/lc3d/Menna_etal_lowcost3d-2012-Trento.pdf [Fecha de consulta: Agosto 2014]
- Software:
 - CloudCompare: <http://www.danielgm.net/cc/> [Fecha de consulta: Julio-Agosto 2014]
 - Geomagic Design Direct: <http://www.geomagic.com/es/products/spark/features/> [Fecha de consulta: Agosto 2014]
 - Polyworks: <http://www.innovmetric.com/en/case-studies/scanning-and-inspection-pty> [Fecha de consulta: Agosto 2014]
 - RealWorks:

- <http://surveying-mapping.duncan-parnell.com/Trimble-RealWorks-pluRealworks.html> [Fecha de consulta: Agosto 2014]
- <http://www.trimble.com/3d-laserscanning/realworks.aspx?dtID=overview&> [Fecha de consulta: Agosto 2014]

8. AGRADECIMIENTOS

En primer lugar, quiero mostrar mi agradecimiento a Pablo Rodríguez González, por la predisposición que ha demostrado en todo momento, atendiendo a las dudas surgidas y ayudando en la captura del brazo robótico.

Del mismo modo, me gustaría agradecer a la Universidad de Salamanca, y en concreto a la Escuela Politécnica Superior de Ávila, el haberme facilitado el instrumental y el software necesarios para llevar a cabo este proyecto.

Por último, dar las gracias al grupo TIDOP por haberme permitido utilizar sus instalaciones durante el trabajo de campo, y haberme ayudado a transportar el motor hasta allí, lo cual no habría sido posible sin ellos.

ANEXOS

ANEXO I:

- Especificaciones técnicas Romer Absolute Arm



ROMER ABSOLUTE ARM CON SEIS EJES DE MOVIMIENTO.

El ROMER Absolute Arm con seis ejes rotacionales está diseñado para mediciones táctiles altamente precisas en incontables piezas de trabajo.

El ROMER Absolute Arm de seis ejes permite la inspección fiable de piezas en las características de piezas de chapa metálica, componentes plásticos o estructuras de fibra de carbono. Si sus trabajos de medición requieren más tarde escaneo láser, es posible la actualización en todo momento.

ROMER ABSOLUTE ARM CON ESCÁNER LÁSER INTEGRADO.

Libertad de movimiento: con un sistema de escaneo láser totalmente integrado y certificado, se trata de un sistema de metrología todoterreno para un amplio espectro de aplicaciones: digitalización 3D, modelado 3D, inspección de nube de puntos, ingeniería inversa, prototipado rápido o fresado de copia son las aplicaciones de escáner láser más frecuentes. El escáner láser es adecuado para una amplia variedad de materiales sin comprometer la precisión. El escáner láser integrado de ROMER no necesita tiempo de calentamiento, cables ni controladores adicionales. El cambio de escaneo a palpado y viceversa es posible en todo momento.



ROMER ABSOLUTE ARM CON ESCÁNER LÁSER EXTERNO.

El ROMER Absolute Arm con escáner externo es una plataforma de escaneo láser de altas prestaciones diseñada para el escáner CMS108 de Hexagon Metrology. Con CMS108, el ROMER Absolute Arm ofrece rendimiento de primera clase incluso en superficies complejas y en piezas de trabajo fabricadas con los tipos de material más difíciles. No se requiere el ajuste del láser conforme al material, ya que el control de potencia láser automático del CMS108 se adapta automáticamente a las condiciones de la superficie. CMS108 es el primer escáner láser con función de zoom, lo que permite tres anchos de líneas diferentes. También pueden conectarse escáneres de otros fabricantes.

SOLUCIÓN DE INSPECCIÓN DE TUBOS ROMER

La solución de inspección de tubos ROMER representa un sistema unificado que cubre las tres principales tareas de medición de tubos: inspección y definición de tubos, medición de geometrías e interconexión de dobladoras con corrección de programa de doblado en línea. El sistema ROMER es la única solución de inspección de tubos portátil del mercado. Puede llevarse hasta la pieza de trabajo para medir tuberías, canalizaciones, mangueras y tubos in situ, ahorrando así tiempo y esfuerzo. La geometría de tubos por ingeniería inversa para piezas de sustitución es rápida y directa, incluso en el lugar de ensamblaje sin necesidad de retirar el tubo.

ROMER ABSOLUTE ARM. ESPECIFICACIONES.

Especificaciones de palpado de 6 ejes

| series 73 | Modelo | Área de medición | Repetibilidad de puntos ¹ | Precisión volumétrica ² | Pesos de brazo |
|-----------|--------|------------------|--------------------------------------|------------------------------------|----------------|
| | 7315 | 1,5 m | 0,025 mm | ± 0,037 mm | 7,1 kg |
| 7320 | 2,0 m | 0,030 mm | ± 0,042 mm | 7,4 kg | |
| 7325 | 2,5 m | 0,038 mm | ± 0,051 mm | 7,7 kg | |
| 7330 | 3,0 m | 0,059 mm | ± 0,075 mm | 8,0 kg | |
| 7335 | 3,5 m | 0,079 mm | ± 0,100 mm | 8,3 kg | |
| 7340 | 4,0 m | 0,099 mm | ± 0,125 mm | 8,6 kg | |
| 7345 | 4,5 m | 0,120 mm | ± 0,150 mm | 8,9 kg | |
| series 75 | 7520 | 2,0 m | 0,016 mm | ± 0,023 mm | 7,7 kg |
| | 7525 | 2,5 m | 0,020 mm | ± 0,029 mm | 8,0 kg |
| | 7530 | 3,0 m | 0,030 mm | ± 0,044 mm | 8,3 kg |
| | 7535 | 3,5 m | 0,040 mm | ± 0,057 mm | 8,6 kg |
| | 7540 | 4,0 m | 0,055 mm | ± 0,069 mm | 8,9 kg |
| | 7545 | 4,5 m | 0,070 mm | ± 0,082 mm | 9,2 kg |

Toda las especificaciones conforme a B89.4.22 y VDI/VDE 2617-9.

Especificaciones de palpado y escaneo de 7 ejes

| series 73 | Modelo ³ | Rango de medición | Repetibilidad de puntos de palpado ¹ | Precisión volumétrica de palpado ² | Sistema de escaneo Precisión SI ⁴ | Sistema de escaneo Precisión SE ⁴ | Pesos del brazo SI | Pesos de brazo SE |
|-----------|---------------------|-------------------|---|---|--|--|--------------------|-------------------|
| | 7320SI/SE | 2,0 m | 0,044 mm | ± 0,061 mm | 0,079 mm | 0,075 mm | 8,3 kg | 7,9 kg |
| 7325SI/SE | 2,5 m | 0,049 mm | ± 0,069 mm | 0,084 mm | 0,080 mm | 8,6 kg | 8,2 kg | |
| 7330SI/SE | 3,0 m | 0,079 mm | ± 0,100 mm | 0,119 mm | 0,113 mm | 8,9 kg | 8,5 kg | |
| 7335SI/SE | 3,5 m | 0,099 mm | ± 0,125 mm | 0,147 mm | 0,140 mm | 9,2 kg | 8,8 kg | |
| 7340SI/SE | 4,0 m | 0,115 mm | ± 0,151 mm | 0,181 mm | 0,172 mm | 9,5 kg | 9,1 kg | |
| 7345SI/SE | 4,5 m | 0,141 mm | ± 0,179 mm | 0,214 mm | 0,203 mm | 9,8 kg | 9,4 kg | |
| series 75 | 7520SI/SE | 2,0 m | 0,023 mm | ± 0,033 mm | 0,058 mm | 0,053 mm | 8,6 kg | 8,2 kg |
| | 7525SI/SE | 2,5 m | 0,027 mm | ± 0,038 mm | 0,063 mm | 0,058 mm | 8,9 kg | 8,5 kg |
| | 7530SI/SE | 3,0 m | 0,042 mm | ± 0,058 mm | 0,083 mm | 0,078 mm | 9,2 kg | 8,8 kg |
| | 7535SI/SE | 3,5 m | 0,055 mm | ± 0,081 mm | 0,101 mm | 0,096 mm | 9,5 kg | 9,1 kg |
| | 7540SI/SE | 4,0 m | 0,067 mm | ± 0,098 mm | 0,119 mm | 0,114 mm | 9,8 kg | 9,4 kg |
| | 7545SI/SE | 4,5 m | 0,084 mm | ± 0,119 mm | 0,138 mm | 0,133 mm | 10,1 kg | 9,7 kg |

Todas las especificaciones en relación a B89.4.22.

ROMER Absolute Arm. Escáneres láser.

| Specifications palpador de scanning | Scanner integrado RS2 | Scanner externo Hexagon CMS108 |
|---------------------------------------|---|-----------------------------------|
| | Velocidad máx. de adquisición de puntos | 50'000 Points/s |
| Puntos por línea | 1000 | max. 2000 |
| Frecuencia de línea | 50 Hz | max. 53 Hz |
| Ancho de línea (val. medio) | 65 mm | 124 mm / 60 mm / 25 mm |
| Distancia (val. medio) | 150 mm ± 50 mm | 180 mm ± 40 mm |
| Espaciado min. de puntos (val. medio) | 0,046 mm | 0,025 mm |
| Control de potencia láser | Semiautomático – por línea | Totalmente automático – por punto |
| Precisión (2 sigma) ⁵ | 30 µm | 20 µm |
| Peso | 340 g | 398 g |
| Unidad de control | No | Si |
| Seguridad láser | Clase 2M | Clase 2 |
| Temperatura de trabajo | 5°C – 40°C (41°F – 104°F) | 10°C – 42°C (50°F – 108°F) |

¹ La Prueba de repetibilidad de puntos es la prueba de referencia para determinar la repetibilidad del brazo de medición con palpador de esfera. El cono está en frente de la máquina. Los puntos se miden desde múltiples direcciones de aproximación. Se calcula el punto promedio y la desviación de cada punto respecto al centro promedio. El resultado es el rango máximo dividido entre dos.

² La Prueba de precisión volumétrica es la prueba más representativa para la precisión de medición de volúmenes. Se mide varias veces y a lo largo de todo el volumen de trabajo un bloque patrón con una longitud conocida y certificada. El bloque patrón se coloca en distintas posiciones y se mide desde múltiples ángulos. El resultado es la desviación máxima de la distancia de medición menos la longitud teórica.

Condiciones ambientales
 Temperatura de trabajo: 0°C – 50°C (32°F – 122°F)
 Temperatura de almacenamiento: -30° – 70° C (-22°F – 158°F)
 Humedad relativa: 10% - 90% sin condensación
 Elevación operacional: 0 – 2000 m (0 – 6600 ft)
Marcas de conformidad
 Conformidad CE: SI

Requisito de potencia
 Voltaje universal 110V – 240V

³ SI designa al ROMER Absolute Arm con escáner integrado. SE designa al ROMER Absolute Arm con escáner externo.

⁴ La Prueba de precisión del sistema de escaneo representa con la máxima precisión las expectativas razonables para el rendimiento de máquina en aplicaciones de medición prácticas mientras se usa el método de escaneo láser. La prueba consiste en la medición de una esfera gris mate con 5 articulaciones de brazo distintas. En cada articulación del brazo, la esfera se escanea desde 5 direcciones distintas, escaneándose así la mayoría de la esfera. El resultado es la distancia máxima centro a centro 3D de las 5 esferas.

Todas las especificaciones de palpado se logran con un ROMER Absolute Arm montado en una base magnética o una placa base ROMER y utilizando un palpador con esfera de acero de 15 mm y 50 mm de longitud bajo condiciones ambientales estables. Todas las especificaciones con el escáner CMS108 se logran con un ROMER Absolute Arm montado en una base magnética o una placa base ROMER y usando el ajuste de zoom 2A del CMS108 (ancho de escaneo de rango medio 60 mm y espaciado de puntos 0,03 mm) y una esfera de calibración gris mate de 25,4 mm de diámetro bajo condiciones ambientales estables.

Todas las especificaciones con el escáner RS2 se logran con un ROMER Absolute Arm montado en una base magnética o una placa base ROMER y una esfera de calibración gris mate de 25,4 mm de diámetro bajo condiciones ambientales estables.



ANEXO II:

- Nube_puntos_Kinect.asc
- Nube_puntos_brazorobótico.asc
- Modelo_poligonal_Kinect.stl
- Modelo_poligonal_brazorobotico.stl

- Nube_individual_vs_Nube_global.bin
- Nube_vs_modelado.bin
- Malla_brazo_vs_Modelado_3D.bin
- Nube_vs_malla_brazo.bin

