

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA POLITECNICA SUPERIOR DE GANDIA

Grado en Ing. Sist. de Telecom., Sonido e Imagen



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCOLA POLITÈCNICA
SUPERIOR DE GANDIA

“Aplicación industrial para el guiado de robots basada en cámaras estéreo 3D”

TRABAJO FINAL DE GRADO

Autor/a:

Carlos Vicente Sáez Rufete

Tutor/a:

Jose Ignacio Herranz Herruzo

Lorena Rausell Curbelo

GANDIA, 2019

“A mi familia
A mis amigos
A mis profesores
A mis compañeros”

Resumen

Este trabajo de fin de grado consiste en la investigación y desarrollo de una aplicación industrial cuya finalidad es devolver una corrección de 6DOF (6 Degrees Of Freedom) con respecto a un sistema de coordenadas global. Estas correcciones serán utilizadas por robots para transformar sus itinerancias de inspección de piezas y poder realizar correctamente su función.

Las correcciones se efectúan mediante un algoritmo conocido como ICP (Iterative Closest Point), que consiste en introducir dos nubes de puntos en 3 dimensiones con las mismas características una trasladada y rotada aleatoriamente con respecto a la otra y, mediante un método iterativo, obtener dicha corrección. Este método calcula la raíz de la distancia media que existe entre los puntos afines de las nubes de puntos al cuadrado RMS (Root Mean Square) en cada iteración y se cerciora de que vaya decreciendo hasta llegar a una convergencia de número de iteraciones máximo o de distancia mínima RMS calculada entre iteraciones.

Palabras clave— *visión artificial, robótica, corrección, coordenadas, ICP*

Abstract

This end-of-degree project consists of investigating and developing an industrial application in order to return a correction of the 6DOF (6 Degrees Of Freedom) according to the global coordinate system. This corrections will be used by robots to transform their roaming of the component inspections and finally do their work properly.

The corrections are carried out by the algorithm known as ICP (Iterative Closest Point) which consist of introducing two three-dimensional point clouds with the same features, one traslated and rotated respect to the other and, through an iterative method, obtain those correction. This method calculates the root mean square (RMS) distance between point clouds and ensures that it decreases untill reaches the maximum number of iterations or the minimum distance RMS calculated each iteration.

Keywords— *artificial vision, robots, correction, coordinates, ICP*



Índice

1	Introducción	2
1.1	Necesidades del cliente	3
1.2	Especificaciones	3
1.3	Solución genérica planteada para satisfacer las necesidades del cliente	4
1.4	Layout de planta	6
2	Objetivos	7
3	Esquema	9
4	Estudio conceptual de posicionamiento 3D	10
5	Trabajos previos de investigación	14
5.1	Evaluación del hardware	14
5.2	Evaluación del software	21
6	Calibración del sensor	26
6.1	Parametrización de la adquisición	28
7	Tratamiento de la información	34
7.1	Parametrización del procesado	39
8	Diseño de la estación de posicionamiento	42
8.1	Placa de calibración	43
8.2	Placa de referencias	44
9	Implantación del sistema	45
9.1	Modular build	45
9.2	Puesta en marcha del sistema	46
10	Evaluación de resultados	47
11	Conclusiones	48

Índice de figuras

1	Piezas para la inspección	4
2	Layout de planta	6
3	Esquema	9



4	Ángulos de Euler	11
5	Modelos N30/N35	14
6	Modelos X30/X36	15
7	Gráficas X36	17
8	Gráficas N35	19
9	Gráficas N30	20
10	Comparativa práctica	21
11	Árbol Octal de una nube de puntos simple	22
12	Diagrama general de comunicación Librería-Programa	23
13	Gráfica de tiempos respecto RMS	24
14	Gráfica de tamaño de la nube respecto RMS	24
15	NxView: Ventana principal	26
16	Patrón de calibración Halcon	27
17	NxView: Calibración	28
18	Profundidad a partir de la disparidad	29
19	NxView: Ventana de parámetros	30
20	Tipos de conectividad	31
21	Relación de singularidad	33
22	Comparativa del procesado de una nube de puntos	34
23	Filtros de procesado de las nubes de puntos	34
24	Interior del filtro de recorte LabVIEW	35
25	Interior del filtro de resamplero LabVIEW	36
26	Interior del filtro de SOR LabVIEW	37
27	Interior de la ICP LabVIEW	38
28	Diseño de la estación ST15	42
29	Diseño de la placa de calibración	43
30	Diseño de la placa de referencias	44
31	Modular Build	46
32	Estación ST15 en puesta en marcha	47

Índice de cuadros

1	Tiempos de ciclo	7
2	Variabilidades en la dejada	8
3	Tabla de valores X36	18
4	Tabla de valores N35	19
5	Tabla de valores N30	20
6	Tabla comparativa de sensores	21
7	Resultados	47

1 Introducci3n

Desde la invenci3n de la cadena de montaje, la industria se ha ido modernizando hasta llegar a un punto en el que no es necesaria la intervenci3n humana. Este cambio ha supuesto un gran avance en las f3bricas que, a la larga, apuestan a favor de estas nuevas tecnolog3as para mejorar tanto eficientemente como eficazmente sus productos. Esto provoca un aumento en la demanda sobre la investigaci3n y el desarrollo de procesados de im3genes para detecci3n de imperfecciones en productos, evitando que salgan a la venta en mal estado, con rasgaduras que impidan el correcto funcionamiento del mismo o que, en general, est3n defectuosos.

En este proyecto, se pretende investigar y desarrollar una de las estaciones de la cadena de montaje que se va a implantar en la f3brica Faurecia Interior Systems S.L. , donde se realizan piezas de inyecci3n de pl3stico para carrocer3as de coches. Estas piezas al ser manipuladas corren el riesgo de golpearse, rallarse, deformarse, romperse o decolorarse entre otros defectos. Por ello, se va a implantar en la f3brica unas estaciones automatizadas que lleven a cabo la identificaci3n y el descarte automatizado de las piezas defectuosas. Pero antes de la identificaci3n de dichos defectos, existe una dificultad previa que se debe solventar. Este problema concreto es el que se va a solventar en este proyecto, el cual proviene de la dejada de las piezas en la banda de transporte que realiza un robot cartesiano. Este robot, al no poder acercarse a la banda de transporte que est3 implantada en nuestro proyecto, se ve obligado a soltar la pieza de manera que la posici3n conocida en la que se encontraba ahora es totalmente aleatoria. Aqu3 es cuando surge la necesidad de implantar una estaci3n previa que calcule cu3nto se ha movido la pieza respecto a una posici3n conocida.

La estaci3n se compone de un sensor marca ENSENSO[®], el cual tiene integradas dos c3maras separadas entre s3 una distancia conocida y una luz azul provista de un patr3n ca3tico. Estos componentes son capaces de construir una nube de puntos en 3 dimensiones obteniendo, por m3todo de disparidad, la profundidad de los objetos y, por m3todo de correlaci3n entre p3xeles afines, la posici3n en el plano XY . Una vez capturadas las nubes de puntos, hay que procesarlas para que devuelvan correcciones de la posici3n con respecto a una de referencia. Este proceso se llevar3 a cabo en la aplicaci3n Visual Studio donde se programar3n en lenguaje C++ unas bibliotecas de enlaces din3micos (ficheros de extensi3n .dll) que utilizar3n la librer3a CCLib encargada de realizar filtros y algoritmos a nubes de puntos. M3s adelante, con la aplicaci3n LabVIEW se conectar3n esos procesados y se obtendr3 una matriz 4x4 llamada matriz de transformaciones donde se encuentran las traslaciones y rotaciones que se deben de aplicar a los robots para corregir la desviaci3n de la pieza.

1.1 Necesidades del cliente

Toda necesidad que se plantea viene dada a un problema pasado, presente o futuro que pueda surgir. En este caso, el cliente se puso en contacto con la empresa AUTIS Ingenieros porque deseaba automatizar un proceso de manipulación de piezas de inyección de plástico e inspección de defectos. Más concretamente, las necesidades del cliente son las siguientes:

- Automatizar el proceso de inspección.
- Automatizar la manipulación de las piezas.
- Clasificar y guardar en una base de datos todas las piezas y sus defectos.
- Disminuir el tiempo de ciclo de todo el proceso de inspección.

Teniendo en cuenta estas necesidades, el proyecto debe avanzar acorde a ellas.

1.2 Especificaciones

Las especificaciones técnicas generales que se deben de tener en cuenta para realizar con éxito el proyecto son las nombradas a continuación.

- La estructura debe ser estable.
- La estructura debe ser de marco soldado mecánicamente y de color blanco (RAL9016).
- La estructura no debe presentar ningún elemento que pueda dañar al operador.
- La estructura debe de tener un acceso seguro para ser mantenida y limpiada sin peligro.
- El sistema debe poder funcionar correctamente en el rango de temperaturas de 8°C a 40°C.
- Las piezas de plástico a procesar son:



Figura 1: Piezas para la inspección

- Las piezas se deben pesar.
- Detección de marcas de expulsión, rayas, off-color, grasas y aceites y todo defecto visible.
- Detección de falta de llenado en las piezas de plástico.
- Rechazo de las piezas de plástico defectuosas.
- Guardado en sacas de las piezas de plástico correctas.
- La banda de transporte debe de colocarse de manera que siga el método de trabajo de los operarios.
- El sistema completo debe cumplir el tiempo de ciclo actual, mejorándolo si es posible.

1.3 Solución genérica planteada para satisfacer las necesidades del cliente

En términos generales, estos son los procesos principales considerados en la presente solución. Se denota cada estación de la celda como ST (Station). Se recomienda observar el Layout de planta para ver mejor la distribución: Figura 2:

Pre-ST10) Las piezas moldeadas salen de las inyectoras y estas son cogidas por un robot cartesiano el cual las deja en la báscula dispuesta en el siguiente punto.

ST10) Después de la dejada de la pieza moldeada, se pesará y se descartará si esta pieza no cumple con el peso establecido. Esta báscula se situará como la primera estación de inspección de la instalación.



- ST15)* Ya pesada, se procederá a comprobar la posición actual de las piezas moldeadas dejadas por el robot cartesiano en el *conveyor*. Esta comprobación es crucial debido a que, en la dejada, las piezas pueden desviarse de la posición teórica inicial causando un funcionamiento incorrecto del sistema. Para evitarlo, una segunda celda de inspección se instalará para detectar automáticamente y establecer la posición nueva de la pieza.
- ST20)* Una vez la posición actual se ha determinado, se procede a la inspección para el control de calidad de la superficie de las piezas moldeadas. Esta se llevará a cabo en la tercera estación de inspección donde serán detectados posibles defectos superficiales.
- ST30)* Una vez inspeccionada superficialmente la pieza, se desplazará hacia la siguiente estación dedicada a la detección de piezas incompletas, es decir, de piezas en las que en algunas zonas no ha llegado el plástico dentro del molde. Esta detección consiste en que un robot coge la pieza desde el *conveyor* y la manipula para que un sensor fijo inspeccione las zonas programadas. A continuación, se lee el *datamatrix* que incorpora la pieza para que se identifique y, por último, el robot puede descartar la pieza dejándola en un *conveyor* o trasladarla a la siguiente estación. Esta estación se localiza dentro del proyecto como la cuarta estación de inspección.
- ST40)* El robot anterior se encarga de posicionar la pieza de forma que el robot de la quinta estación de inspección pueda cogerla adecuadamente y trasladarla hacia las sacas donde se van a almacenar. Esta parte es muy delicada puesto que las sacas donde se almacenan se tienen que controlar minuciosamente y el material del que están hechas no es muy rígido.
- ST50)* Si durante todo el proceso de inspección de las piezas se ha detectado algún defecto, el robot de la estación ST30 pasará a esta sexta y última estación, que transportará la pieza fuera de la celda y se rechazará.

1.4 Layout de planta

El layout de planta se crea partiendo de los requisitos del cliente y sus especificaciones. En la Figura 2 se muestra el layout visto desde arriba. El texto en violeta indica el nombre de la estación.

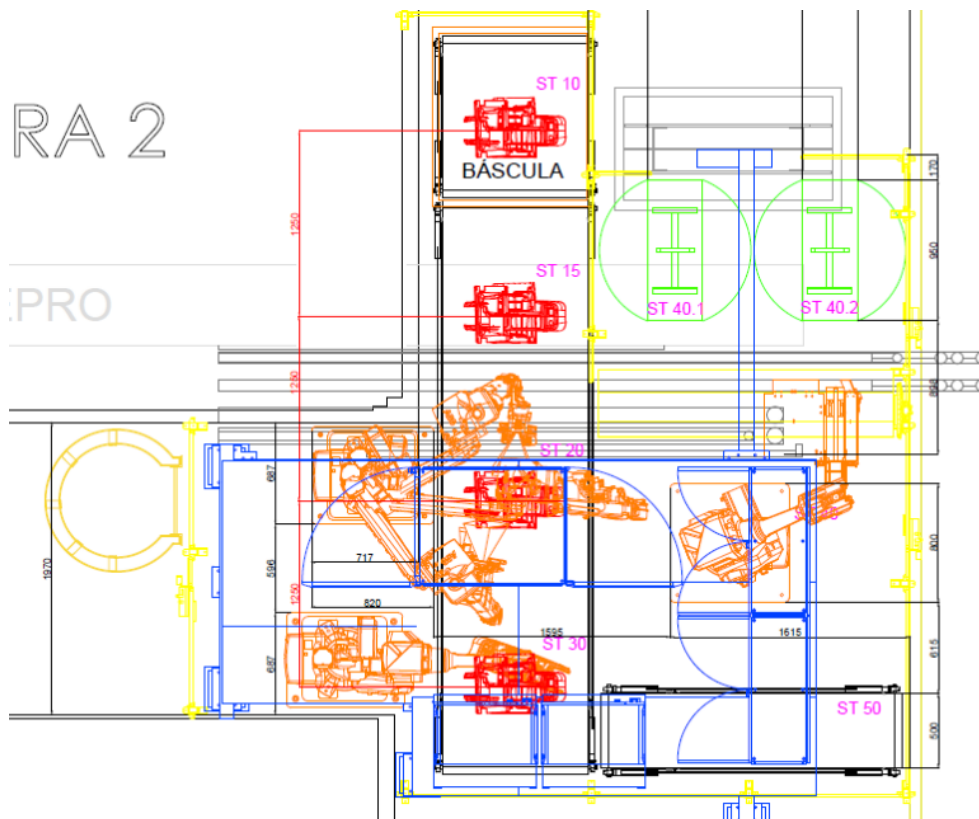


Figura 2: Layout de planta

Se puede ver en la Figura 2 distintos colores asociados a cada una de las clases de objetos que se van a encontrar en el proyecto. El rojo representa el conjunto de piezas dispuesto en cada una de las estaciones; el naranja representa los robots; el azul hace referencia a una plataforma situada sobre el suelo donde se encuentran todos los armarios de control y alimentación de la celda, el rack computacional y la estación de trabajo donde se guardarán y manipularán los datos; el amarillo es el vallado obligatorio para esta clase de proyectos, donde los robots no son colaborativos; el verde, son las sacas donde se insertarán las piezas no defectuosas; y el negro finalmente son las bandas de transporte.

2 Objetivos

Los objetivos de este proyecto son el estudio e incorporación de un sensor estéreo 3D a la estación de inspección *ST15* y el estudio y desarrollo del software encargado del procesamiento de las capturas de dicho sensor. Y, para llegar a cumplir estos objetivos, es necesario que se tenga en cuenta la finalidad de esta estación y las necesidades de su implantación en este proyecto. Las necesidades vienen producidas por el robot encargado de dejar las piezas en el proceso de inspección. Este robot, llamado SEPRO, tiene dos inconvenientes que se deben tener en cuenta y cumplir en la estación *ST15*: el tiempo de ciclo y la variabilidad.

Se denomina al tiempo de ciclo como el tiempo que tarda el robot SEPRO entre que deja una pieza y la siguiente. Este tiempo es muy importante que en cada estación del proyecto no se sobrepase porque si no, se tendrá una demora en las dejadas de las piezas que conlleven un mayor tiempo de ciclo por pieza. Para el cliente es de vital importancia que no ocurra esto porque se disminuye en gran medida la producción. En el Cuadro 1 se muestra la tabla de tiempos de ciclo por pieza.

Pieza	Tiempo Ciclo [s]
Glove Box	55
Glove Box Open	44
Consolas	52
Door Outer	44

Cuadro 1: Tiempos de ciclo

Pero no solamente hay que tener en cuenta el tiempo de ciclo. La variabilidad de las piezas en la dejada que realiza el SEPRO también es algo a tener en cuenta. Esta variabilidad proviene de que el robot, cuando se dispone a dejar una pieza en la celda de inspección, no llega a alcanzar la banda de transporte, por lo que se ve obligado a soltar la pieza desde una altura considerable. Una vez el SEPRO suelta la pieza, se deposita en la banda de transporte de manera impredecible y aleatoria. Mediante un estudio previo se determinó la variabilidad máxima que puede tomar cada pieza dentro de esa aleatoriedad. Esto nos permite medir el máximo de desplazamiento y rotación que se obtienen en las dejadas y tenerlo en cuenta a la hora de procesar la pieza. Por lo tanto, la estación *ST15* tendrá que poder corregir piezas con las variabilidades impuestas por la caída de las piezas, las cuales se observan en el Cuadro 2.



Modelo	Variabilidad ψ [grados]	Variabilidad x [mm]	Variabilidad y [mm]
Glove Box Open	5	10	30
Glove Box	15	40	40
Door Outer	5	10	30
Consola	10	20	20

Cuadro 2: Variabilidades en la dejada

Si la estación *ST15* es capaz de cumplir el tiempo de ciclo y corregir las piezas dentro de la variabilidad establecida, se podrá decir que se cumplen los objetivos de estudio y desarrollo establecidos en este proyecto final de grado.

3 Esquema

Para cumplir los objetivos nombrados anteriormente, se ha decidido realizar un esquema organizativo para un mejor entendimiento del desarrollo y de los pasos a realizar en este proyecto. Para ello, se van a enumerar los puntos de este proyecto y se dará una breve explicación de ellos apoyándose en un esquema por bloques.

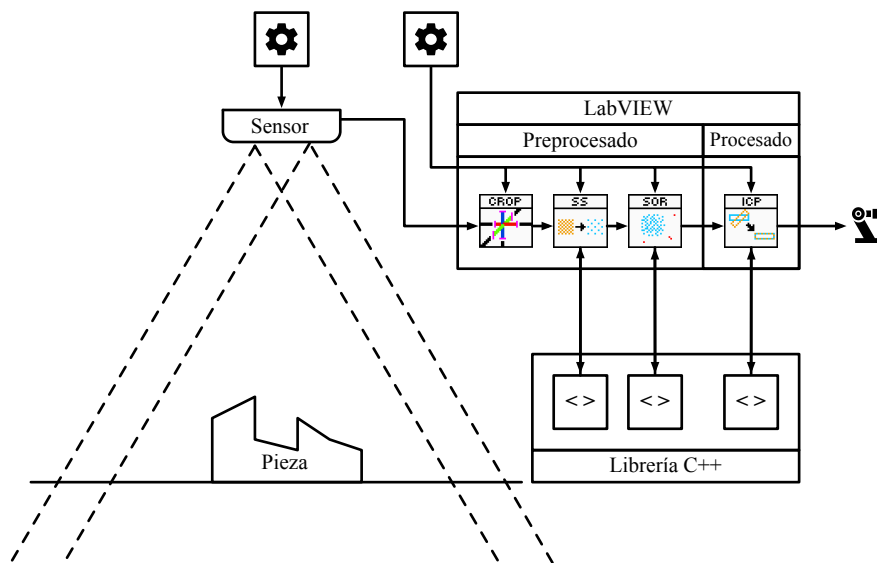


Figura 3: Esquema

1. Un sensor capaz de capturar nubes de puntos es de vital importancia para la estación *ST15*. Se tendrá que escoger uno del mercado que ofrezca las mejores características y se amolde al proyecto lo mejor posible (Sección 5.1) y se calibrará para asegurarse de su buen funcionamiento (Sección 6).
2. Una vez obtenida la nube de puntos, se tiene que procesar. Para ello, se deben buscar las herramientas necesarias (Sección 5.2) y adecuarlas a nuestro sistema (Sección 7).
3. Y a la salida de nuestra estación se tendrán unas correcciones del sistema de coordenadas de los robots de las estaciones posteriores (Sección 4).

4 Estudio conceptual de posicionamiento 3D

Los robots dispuestos en las celdas de inspección tienen 6 grados de libertad (6DOF: *6 Degrees Of Freedom*), es decir, traslaciones X , Y y Z y rotaciones ϕ , ψ y θ , en los cuales se han de aplicar correcciones. Esas correcciones se pueden aplicar gracias a las transformaciones afines.

En álgebra lineal, las transformaciones lineales son representadas con matrices. Si T es una transformación lineal asignada de \mathbb{R}^n a \mathbb{R}^m siendo \vec{x} el vector columna con n entradas, entonces

$$T(\vec{x}) = A\vec{x} \quad (1)$$

para alguna $m \times n$ matriz A , llamada matriz de transformación de T . Nótese que A tiene m filas y n columnas, mientras que la transformación T es desde \mathbb{R}^n a \mathbb{R}^m .

Los robots que se encuentran en las celdas son robots cartesianos con 6 grados de libertad (6DOF). Esto significa que las correcciones se deben hacer teniendo en cuenta el orden y el tipo de representación que requieren. Partiendo de que las traslaciones no son algo en lo que preocuparse puesto que no tienen otra forma de representarse, se han centrado los esfuerzos en la representación de los ángulos. Estos ángulos son los llamados ángulos de Euler y también vienen representados en una matriz de 3×3 . Esta matriz contiene las tres variaciones angulares que se aplicarán:

- α es el ángulo entre el eje x y la línea de nodos N o Norte.
- β es el ángulo entre el eje z y el eje Z .
- γ es el ángulo entre la línea de nodos y el eje X .

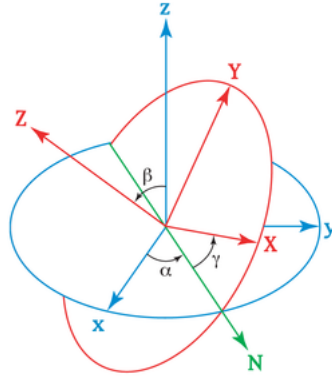


Figura 4: Ángulos de Euler

También se considera la notación $\alpha = \phi, \beta = \psi, \gamma = \theta$, más comúnmente conocidos en robótica como *yaw*, *pitch* y *roll*, respectivamente [1]. Se ha optado por esta nomenclatura.

Hay que tener en cuenta que las matrices no cumplen la propiedad conmutativa y, por tanto, el orden en que se haga la multiplicación afecta al resultado. A continuación, se muestran las matrices correspondientes con cada rotación en cada coordenada:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} R_y(\psi) = \begin{bmatrix} \cos \psi & 0 & \sin \psi \\ 0 & 1 & 0 \\ -\sin \psi & 0 & \cos \psi \end{bmatrix} R_z(\phi) = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$R(\theta, \psi, \phi) = R_x(\theta)R_y(\psi)R_z(\phi) = \quad (3)$$

$$= \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi \sin \phi - \sin \theta \cos \phi & \cos \theta \sin \psi \sin \phi + \sin \theta \sin \phi \\ \sin \theta \cos \psi & \sin \theta \sin \psi \sin \phi + \cos \theta \cos \phi & \sin \theta \sin \psi \sin \phi - \cos \theta \sin \phi \\ -\sin \psi & \cos \psi \sin \phi & \cos \psi \cos \phi \end{bmatrix} \quad (4)$$

Esta matriz $R(\theta, \psi, \phi)$ junto con el vector de traslaciones, forman la matriz de transformación A nombrada anteriormente. Esta matriz al aplicarla a todos los puntos de la nube, varía sus coordenadas tanto en rotación como en traslación:

$$A = \begin{bmatrix} R(\theta, \psi, \phi) & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

, donde

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (6)$$

Una vez explicado el concepto de matriz de transformación, se va a introducir el mismo en el uso del algoritmo llamado ICP (*Iterative Closest Point*) encargado de realizar el cálculo de la matriz de transformación mediante un método iterativo.

El algoritmo ICP es un método estudiado por Paul J. Besl y Neil D. McKay publicado en Febrero de 1992 en un documento llamado 'A Method for Registration of 3-D Shapes' [2]. En este documento se describe a la ICP como un método de registro de figuras en 3 dimensiones incluyendo curvas de forma libre y superficies. Este método requiere de un procedimiento para encontrar el punto más cercano en una entidad geométrica partiendo de otro punto dado. El algoritmo siempre converge al mínimo local más cercano de la distancia métrica al cuadrado. Por consiguiente, dado un conjunto adecuado de muestras con unas rotaciones y traslaciones de una clase particular de objetos con un cierto nivel de complejidad, uno puede globalmente minimizar la distancia media al cuadrado RMSD (*Root Mean Square Deviation*) en los 6DOF. Por ejemplo, una nube de puntos M se ha movido (registrado, posicionado) con respecto a otra nube de puntos equivalente R . Partiendo del ejemplo, el procedimiento iterativo para converger en un mínimo local es:

1. $\forall m \in M$ encuentra el punto más cercano $r \in R$.
2. Transforma $M_{k+1} \leftarrow Q(M_k)$ para minimizar la raíz del error cuadrático medio RMSD entre cada m y r .

$$RMSD(M, R) = \sqrt{\frac{\sum_{i=1}^N \sqrt{(x_{m,i} - x_{r,i})^2 + (y_{m,i} - y_{r,i})^2 + (z_{m,i} - z_{r,i})^2}}{N}} \quad (7)$$

3. Termina cuando se llega a un error mínimo entre iteraciones o cuando se alcanza un número máximo de iteraciones.

Una vez ha llegado a converger la ICP, el método devuelve como resultado la matriz de transformación con la cual se puede corregir M para que concuerde con la nube de puntos de referencia R .

Sin embargo, en todos estos procesos teóricos no se han tenido en cuenta ni el tiempo de ciclo ni el ruido con el que se ven afectadas las capturas de nubes de puntos. Más adelante se



tratará el problema del ruido en las capturas en la Sección 7 y la eficiencia de este algoritmo para reducir el tiempo hasta cumplir con el tiempo de ciclo en la Sección 5.

5 Trabajos previos de investigación

5.1 Evaluación del hardware

La necesidad de que la ICP dure segundos en vez de minutos es crucial para que se escoja esta tecnología, puesto que el cliente ha destacado que no se debe aumentar el tiempo de ciclo, ya que supone un retraso en todo el proceso que deriva en pérdida de ganancias. Por ello, se ha de investigar todas las herramientas que estén a nuestro alcance y desarrollar la que demuestre un mejor rendimiento y eficacia a la hora de su implantación en el proyecto. Esta investigación se divide en dos partes bien diferenciadas: hardware y software.

La adquisición de nubes de puntos es una ardua tarea que requiere de herramientas sofisticadas que consigan la mejor de las capturas adecuándose al presupuesto, área de trabajo y condiciones del entorno acordados en el proyecto. Observando el diseño de la estación de posicionamiento en la Sección 8, se buscan los sensores capaces de cumplir con los requisitos del proyecto. En este caso todos los escogidos para la evaluación son de la marca ENSENSO[®]. Esta marca diseña cámaras estéreo 3D para aplicaciones industriales en el dominio de los alimentos, logística y farmacología. El funcionamiento de estos sensores siempre sigue el mismo principio. Consiste en dos cámaras separadas una distancia conocida y una luz emisora de patrones. Este patrón proyectado es capturado por las dos cámaras, las cuales gracias a su distancia entre ellas y la identificación de las propiedades de ese patrón que según la distancia a la cámara varía, es capaz de obtener una nube de puntos en tres dimensiones. En comparación con otras técnicas de escaneado 3D, este tipo de sensores son compactos y sin ninguna pieza móvil. Esto simplifica la integración en un entorno de fabricación y hace factible su utilización en aplicaciones robóticas.

Una vez escogido el tipo de sensor, se ha de elegir el modelo que cumpla con los requisitos técnicos adecuados [3]. Se disponen de dos tipos de serie de modelos: la serie N y la serie X. La serie N son sensores de una sola pieza compactos provistos de una pre-calibración intrínseca no modificable:

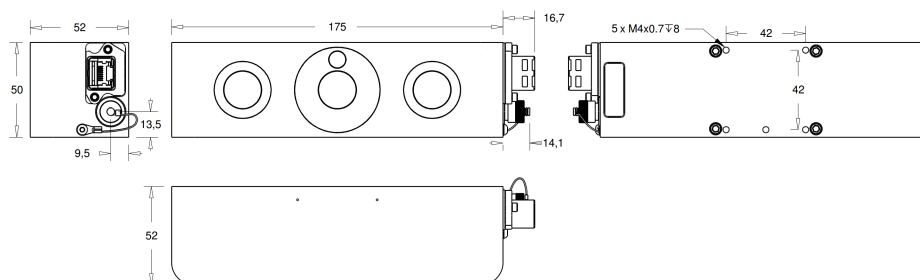


Figura 5: Modelos N30/N35

Mientras que la serie X se caracteriza por ser modificable en términos de calibración intrínseca e intercambio de piezas [4]:

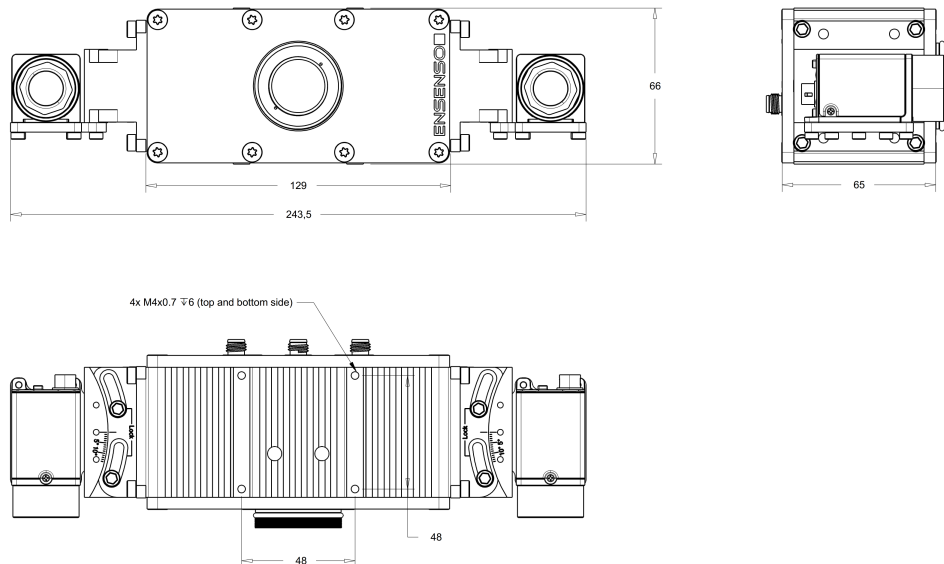


Figura 6: Modelos X30/X36

Teniendo siempre en cuenta las condiciones ambientales de cada uno que se nombran a continuación.

- Temperatura: 0 - 40°C
- Humedad relativa (varía según el modelo):
 - 20 - 80 % (sin condensar) para N10, N20, X30-xCP, X36-xCP
 - 0 - 100 % (sin condensar) para N30, N35, X30-xFA, X36-xFA
- Desviación relativa en Z (varía según la distancia de trabajo):
 - $\leq 5e-5$ / °C a 1400mm
 - $\leq 2.5e-5$ / °C a 300mm
- Desviación angular en X: $\leq 2e-5$ rad / °C
- Desviación angular en Y: $\leq 5e-5$ rad / °C

Considerando la temperatura más alta de trabajo que se facilita en las especificaciones y la distancia de trabajo en este proyecto, la desviación que se debe de esperar es la siguiente:

- $Z_{desv} \leq 4,32e - 5/^{\circ}C \cdot 1100mm \cdot 40^{\circ}C = 1,901mm$
- $X_{desv} \leq 1100mm \cdot \tan(2e - 5/^{\circ}C \cdot 40^{\circ}C) = 0,153mm$
- $Y_{desv} \leq 1100mm \cdot \tan(5e - 5/^{\circ}C \cdot 40^{\circ}C) = 0,384mm$

También se tiene que asegurar que los sensores trabajarán correctamente en situaciones adversas como vibraciones o shocks. Si estas dos situaciones se producen por algún robot cercano por ejemplo, hay que asegurarse que previamente el sensor cumple con los estándares adecuados. En este caso, se debe cumplir el estandar de vibración sinusoidal IEC 60068-2-6 y el de shock IEC 60068-2-27. ENSENSO[®] afirma cumplir estos estándares para la serie N y X a 30-500Hz / 10g en vibración sinusoidal y a 80g/1.9ms, 25g/6ms en shock (Dependiendo de en la serie X qué piezas se le incorpore, se obtendrán estos o peores resultados).

Llegado este punto, se toma una primera conclusión de que las características del producto son más que viables para introducirlo en el proyecto. Por tanto, se contacta con una empresa proveedora de este tipo de sensores industriales pensados para cumplir todas estas condiciones de manera repetitiva y a largo plazo. Como las dos series N y X tienen la misma funcionalidad, es necesario realizar un estudio previo con los modelos que se comporten mejor en el proyecto. Se escogieron tres modelos para este estudio:

- **X36-5CP-8/16/1400-400/1600:** Este modelo está equipado con una entrada conmutador Ethernet Gigabit y puede combinarse con cámaras estéreo 3D con sensores GigE uEye. Tiene una resolución de 1280x1024 píxeles a 1/1.8'' o 2456x2054 píxeles a 2/3'' con sensores Global Shutter CMOS. El proyector adicionalmente contiene la tecnología FlexView2. El aislamiento y conexionado son a prueba de polvo y agua conforme a IP65 y IP67.

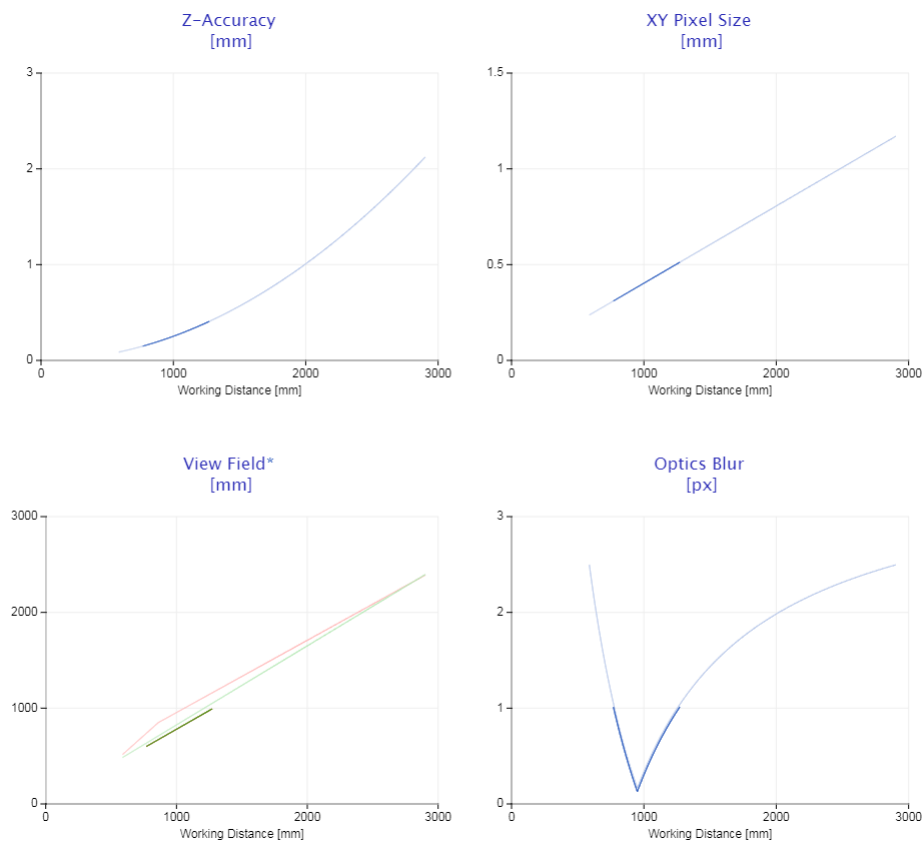


Figura 7: Gráficas X36

En la Figura 7 se puede visualizar el comportamiento del modelo X36 según la distancia a la que se coloque el sensor de los objetos a capturar [4]. Como se puede apreciar en la Figura 28, el área de trabajo se encuentra entre los 600mm y los 1100mm de distancia al sensor. Si observamos la gráfica de *Optics Blur* (Desenfoque óptico), el pico de nivel más bajo de desenfoque óptico se encuentra entre los valores del área de trabajo. Por tanto, el sensor se encuentra en la posición correcta. En el Cuadro 3 se muestran los valores de las gráficas. En gris claro se muestran los valores que se obtienen a lo largo de toda el área de trabajo y en gris oscuro la mejor distancia de trabajo posible con este sensor.

Distancia de trabajo [mm]	Precisión en Z [mm]	Tamaño Pixel [mm]	Campo de visión X [mm]	Campo de visión Y [mm]	Desenfoque óptico [px]
800	0.161	0.322	623.88	624.20	0.833
850	0.182	0.343	662.87	663.21	0.571
900	0.204	0.363	701.86	702.23	0.337
950	0.227	0.383	740.86	741.24	0.129
1000	0.252	0.403	779.85	780.25	0.302
1050	0.278	0.423	818.84	819.26	0.461
1100	0.305	0.443	857.83	858.28	0.605
1150	0.333	0.463	896.83	897.29	0.736
1200	0.363	0.484	935.82	936.30	0.857
1250	0.394	0.504	974.81	975.31	0.968
1272	0.408	0.513	991.97	992.48	1.014

Cuadro 3: Tabla de valores X36

- **N35-602-16-BL:** Los modelos N30 y N35 son los modelos con mayor precisión de la serie N compacta. Tiene conexión Gigabit Ethernet con soporte PoE y un LED de alta intensidad proyector de patrones. Tiene una resolución de 1280x1024 píxeles a 1/1.8'' con sensores Global Shutter CMOS. Adicionalmente, el modelo N35 está equipado con la técnica FlexView. En encapsulado de estos sensores están evaluados en IP65, IP67 y a prueba de agua y polvo.

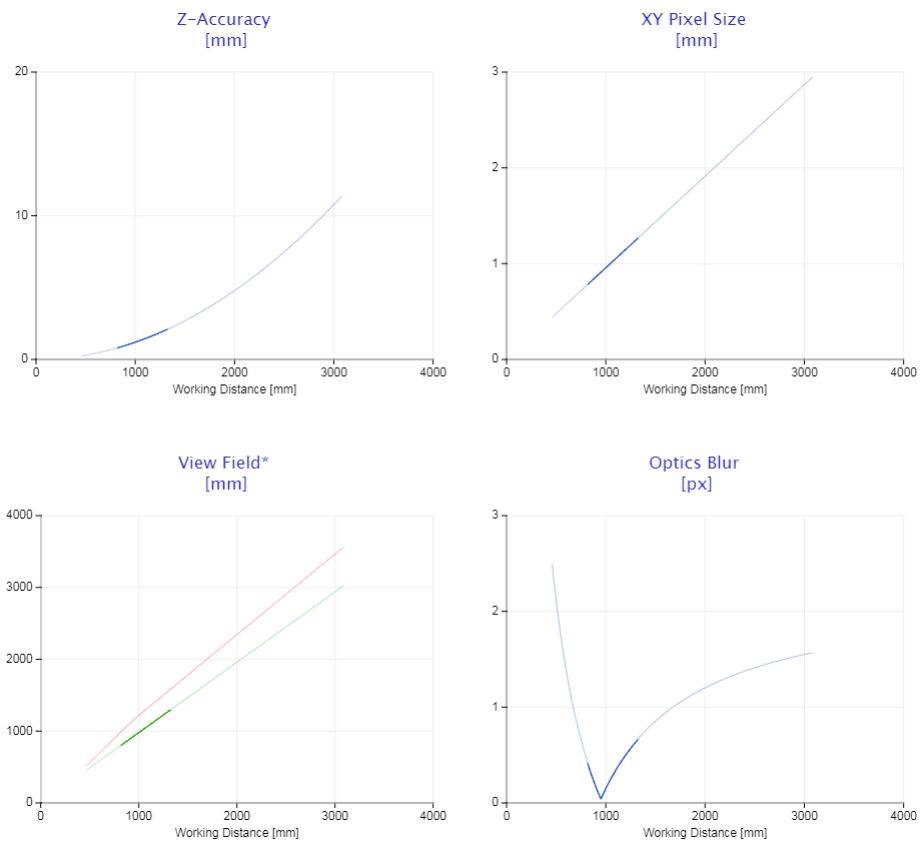


Figura 8: Gráficas N35

Distancia de trabajo [mm]	Precisión en Z [mm]	Tamaño Pixel [mm]	Campo de visión X [mm]	Campo de visión Y [mm]	Desenfoque óptico [px]
850	0.865	0.813	832.60	832.60	0.307
900	0.969	0.861	881.58	881.58	0.166
950	1.080	0.909	930.55	930.55	0.040
1000	1.197	0.957	979.53	979.53	0.150
1050	1.320	1.004	1028.51	1028.51	0.250
1100	1.448	1.052	1077.48	1077.48	0.340
1150	1.583	1.100	1126.46	1126.46	0.423
1200	1.724	1.148	1175.44	1175.44	0.499
1250	1.870	1.196	1224.41	1224.41	0.568
1300	2.023	1.244	1273.39	1273.39	0.633
1324	2.098	1.267	1296.90	1296.90	0.662

Cuadro 4: Tabla de valores N35

- N30-602-16-IR

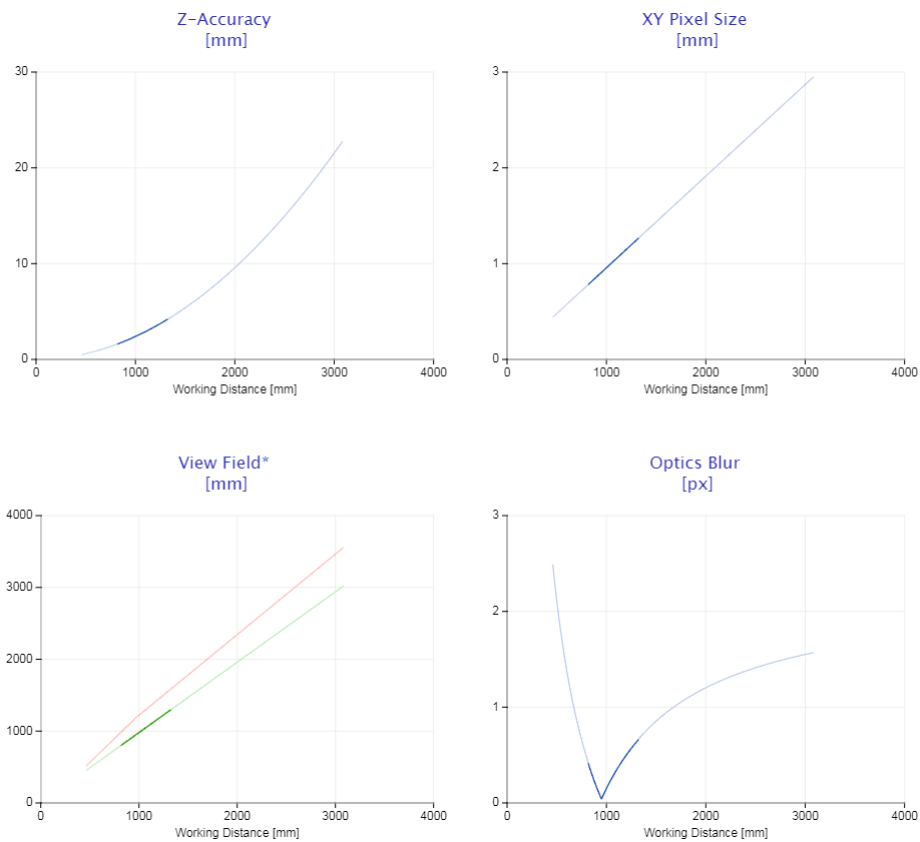


Figura 9: Gráficas N30

Distancia de trabajo [mm]	Precisión en Z [mm]	Tamaño Pixel [mm]	Campo de visión X [mm]	Campo de visión Y [mm]	Desenfoque óptico [px]
850	1.729	0.813	832.60	832.60	0.307
900	1.939	0.861	881.58	881.58	0.166
950	2.160	0.909	930.55	930.55	0.040
1000	2.394	0.957	979.53	979.53	0.150
1050	2.639	1.004	1028.51	1028.51	0.250
1100	2.896	1.052	1077.48	1077.48	0.340
1150	3.166	1.100	1126.46	1126.46	0.423
1200	3.447	1.148	1175.44	1175.44	0.499
1250	3.740	1.196	1224.41	1224.41	0.568
1300	4.045	1.244	1273.39	1273.39	0.633
1324	4.196	1.267	1296.90	1296.90	0.662

Cuadro 5: Tabla de valores N30

En el Cuadro 6, se presenta la tabla comparativa entre los tres modelos añadiendo el presupuesto de cada uno.

Modelo	Máxima precisión en Z [mm]	Precio comparativo
N30	2.160	x
N35	1.080	$1,05x$
X36	0.227	$1,35x$

Cuadro 6: Tabla comparativa de sensores

Para una mejor visualización, se ha capturado la misma pieza en las mejores condiciones y características de cada uno de los modelos. En la Figura 10, se puede apreciar el gran cambio que suponen los tres modelos por tener una menor precisión en Z. También destacar que, de estos tres modelos, el único que no incluye la tecnología FlexView es el N30. Esta tecnología influye mucho en la precisión que puede llegar a tener el sensor pero también en el tiempo que se tarda en capturar una nube de puntos.

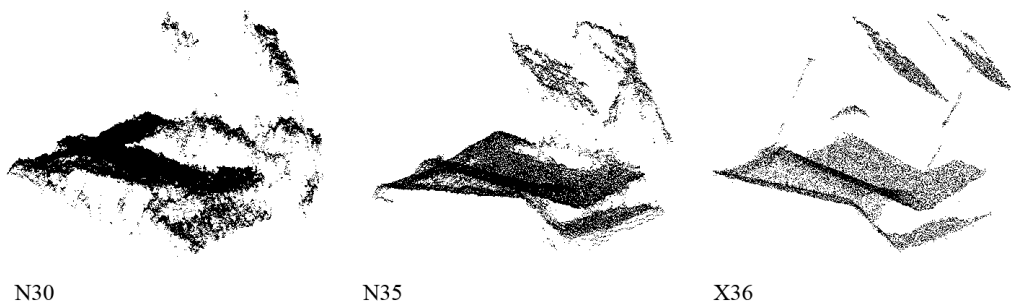


Figura 10: Comparativa práctica

Una vez acabada la evaluación teórica y práctica del hardware, se llega a la conclusión de que el modelo más apropiado para este proyecto es el N35-602-BL, puesto que su precisión en Z es más que suficiente para el trabajo que va a llevar a cabo (no como el N30-602-16-IR) y su precio no es tan elevado como el de su modelo de serie superior X36-5CP-8/16/1400-400/1600. Con el modelo escogido se harán las capturas de nubes de puntos que se procesarán más adelante mediante software, del cual se va a hablar a continuación.

5.2 Evaluación del software

Existen librerías compiladas en C++ de código abierto que suponen una gran ventaja para usuarios que pretendan realizar procesados de nubes de puntos en 3 dimensiones. Una de las librerías más extendidas es PCL (Point Cloud Library). Esta librería está desarrollada por

científicos e ingenieros de empresas de renombre como NVIDIA®, TOYOTA® o universidades como la Carlos III de Madrid. Es una librería bastante extensa provista de un gran número de algoritmos de filtrado, extracción de características, reconstrucción de superficies, registro (ICP), ajustes de modelos y segmentación [5].

Partiendo de PCL, existen otras muchas librerías de código abierto que son más específicas y vienen ya tratadas para que el procesado de las nubes de puntos sea lo más eficiente posible. Es el caso de CCLib. CCLib es una librería desarrollada enteramente por D. Girardeau-Montaut y diseñada para funcionar en un programa también gratuito llamado CloudCompare. CloudCompare es un software libre de procesado de nubes de puntos en 3 dimensiones. Se diseñó originalmente para realizar comparaciones entre dos nubes de puntos densas o entre una nube de puntos y una malla triangular [6]. Esta comparación depende de una estructura llamada árbol octal (Octree). Un árbol octal es una estructura en árbol de datos en la cual cada nodo interno tiene exactamente 8 subnodos. Estas estructuras se usan mayormente para particionar un espacio tridimensional, dividiéndolo recursivamente en ocho octantes disminuyendo notablemente el tiempo de procesado [7].

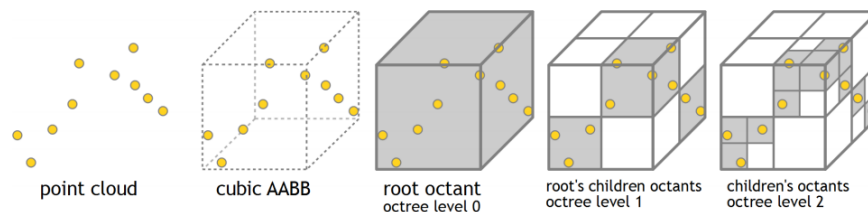


Figura 11: Árbol Octal de una nube de puntos simple

Ahora bien, para poder utilizar dichas librerías en el proyecto hacen falta una serie de conceptos que hay que asimilar antes. Estas librerías están compiladas en C++ y cualquier código que queramos realizar para llamarla tiene que estar en el mismo lenguaje. Por ello, si se pretende llamar a dicha librería desde un programa externo (en este caso LabVIEW) es necesario un contenedor o "wrapper" que traduzca las llamadas desde un programa a llamadas propias del lenguaje de la librería. Ese contenedor se compila en formato .dll (biblioteca de vínculos dinámicos) y finalmente se ejecuta.

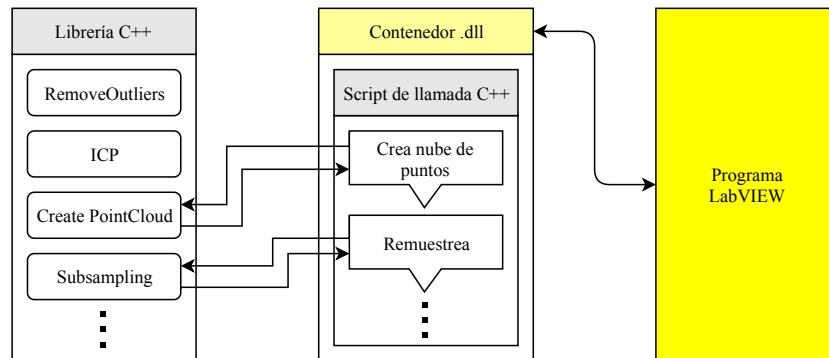


Figura 12: Diagrama general de comunicación Librería-Programa

Si bien PCL y CCLib incorporan el método de la ICP, no se consiguen los mismos resultados a efectos prácticos. A continuación, se muestra el estudio comparativo entre las dos librerías realizado con muestras reales.

A efectos de tiempo de procesado, como se puede observar en la Figura 13, PCL ofrece los mismos resultados que CCLib en tiempos más cortos. Sin embargo, la dispersión de las pruebas es mucho mayor que en el caso de CCLib y esto puede conllevar problemas más adelante ya que el funcionamiento no es similar de una prueba a otra, es decir, no hay repetitividad. Por necesidad, el código ha de ser robusto y, en este caso, CCLib aporta una mayor similitud entre pruebas y, por tanto, robustez.

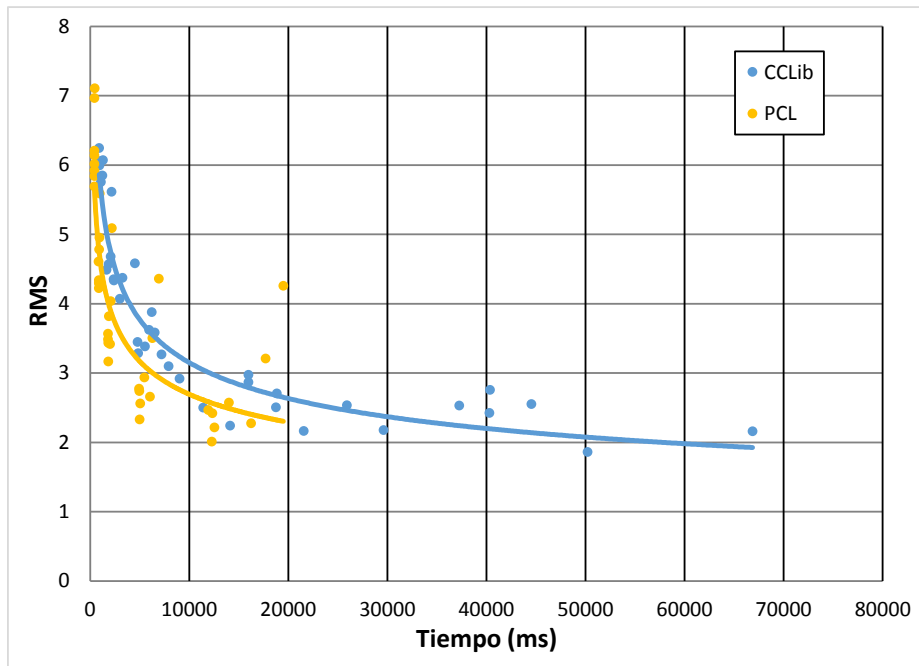


Figura 13: Gráfica de tiempos respecto RMS

A efectos de tamaño de las capturas, CCLib ofrece mejores resultados que PCL puesto que se consigue un menor error con el mismo número de puntos en las muestras.

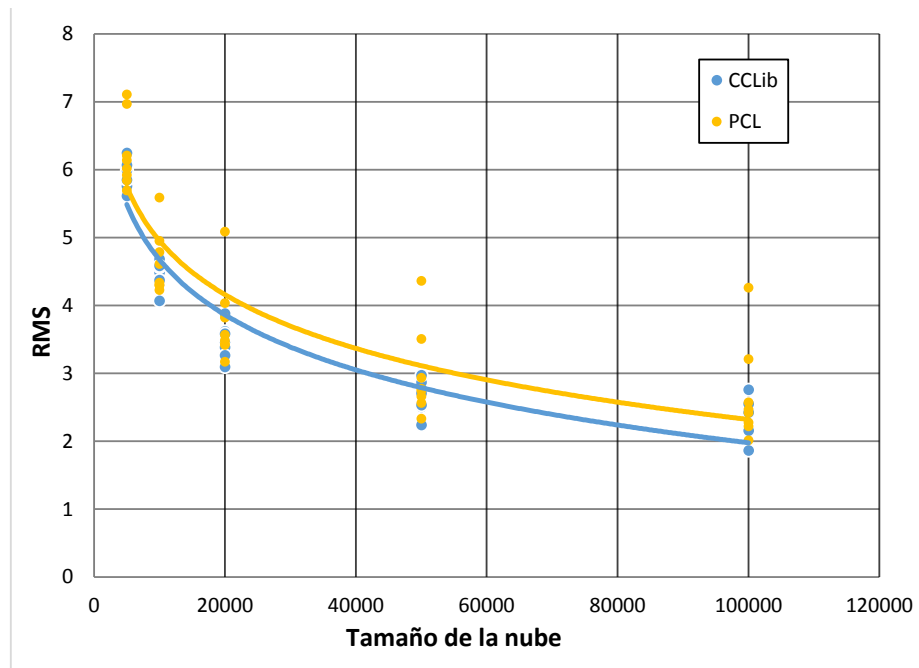


Figura 14: Gráfica de tamaño de la nube respecto RMS



Una vez acabada la evaluación teórica y práctica del software, se llega a la conclusión de que la librería más apropiada para este proyecto es CCLib, puesto que es una librería preparada y enfocada para este tipo de uso. PCL al ser tan extensa, no se centra en lo que verdaderamente se busca en este proyecto.

6 Calibración del sensor

El sensor escogido para este proyecto es un modelo encapsulado que ENSENSO[®] nos dispone calibrado intrínsecamente. La calibración intrínseca se refiere a que la estructura interna está configurada de tal forma que su enfoque o distancia entre las cámaras que componen el sensor no se pueden manipular. Por lo tanto, este sensor sólo puede trabajar correctamente en el área de trabajo especificado en su datasheet. Ahora bien, es necesario que se calibre de manera extrínseca. Esta calibración te permite averiguar cuántos milímetros son un píxel capturado por el sensor y, además, añadir un offset al origen de coordenadas que será necesario para declarar el mismo origen de coordenadas que los robots de las estaciones posteriores.

ENSENSO[®] ofrece un paquete de herramientas de calibración de sus sensores llamado EnsensoSDK, donde se encuentra el programa NxView. Este programa sirve para mostrar en directo el campo de profundidad de la imagen adquirida, calibrar de manera extrínseca y parametrizar la adquisición de datos, entre otros.

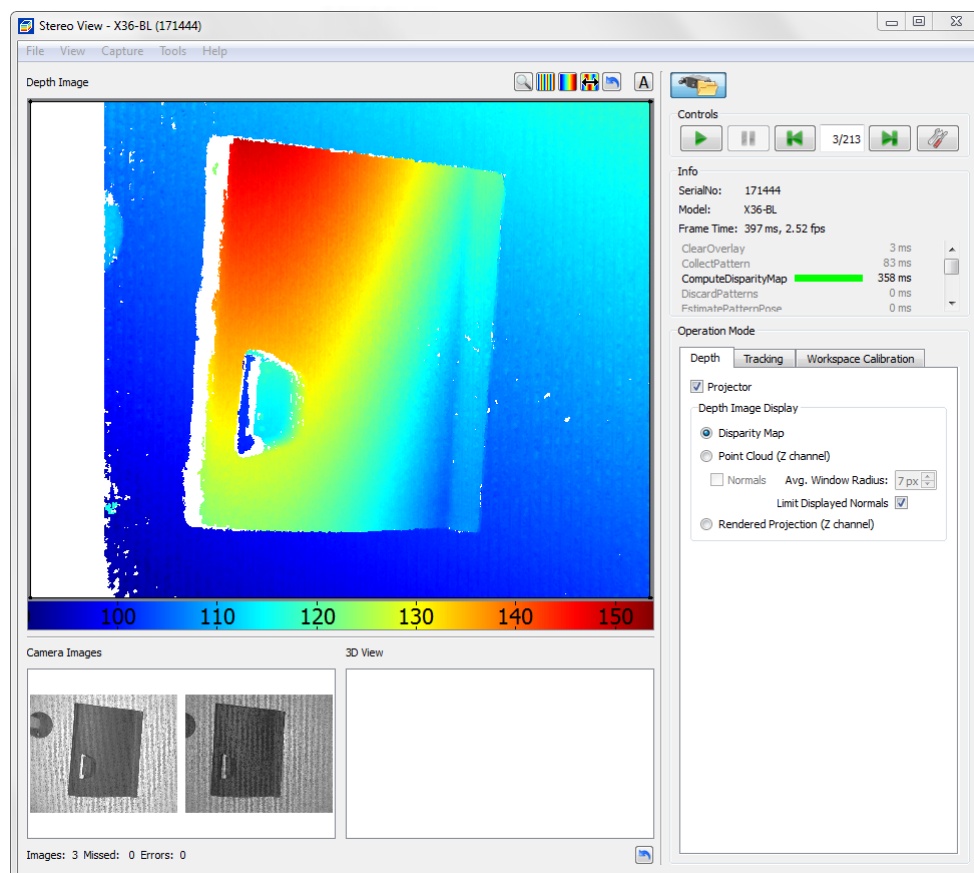


Figura 15: NxView: Ventana principal

Antes de adquirir con el sensor, primero se debe calibrar mediante este programa y un patrón de calibración Halcon.

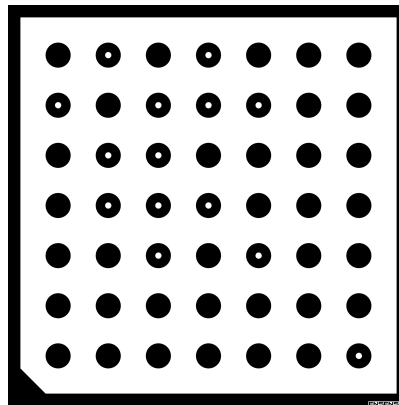


Figura 16: Patrón de calibración Halcon

Este patrón marca en su centro el origen de coordenadas que tendrá el sensor y está codificado de tal manera que los puntos con los centros en blanco marcan el espaciado en mm entre centros de los puntos para realizar la transformación de los píxeles en milímetros. Una vez identificado el patrón, se colocará de manera que se observe completamente por el sensor.

A diferencia del sensor, el origen de coordenadas de los robots se tiene que calibrar mediante un saliente acabado en punta. Este saliente será entonces el origen de coordenadas de los robots que debe ser el mismo que el del sensor. Para ello, mediante una herramienta de NxView, se le tendrá que añadir al origen de coordenadas, situado en el centro del patrón de calibración, un offset dependiendo de dónde se sitúe el origen de los robots.

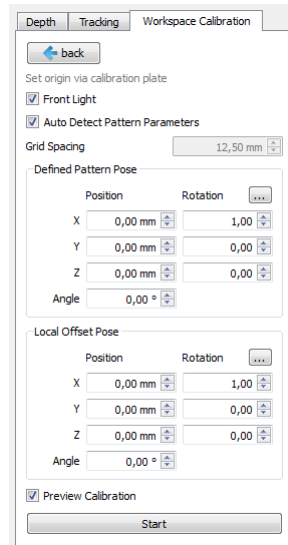


Figura 17: NxView: Calibración

En la Figura 17 se muestra la calibración del sensor mediante el patrón. Se puede observar unos campos a completar para añadir offsets, requeridos más adelante para disponer el origen de coordenadas del sensor en el mismo lugar que el origen de coordenadas de los robots.

6.1 Parametrización de la adquisición

Una vez calibrado el sensor, se debe parametrizar la adquisición de la nube de puntos mediante cambios en la configuración de las cámaras y cambios en la algoritmia encargada de realizar el mapa de profundidad. Esta algoritmia se denomina disparidad. Esta disparidad necesita de dos cámaras de características similares, como las internas del sensor. Estas dos cámaras forman un par estéreo, y cada una de ellas cumplen un modelo *pinhole*. Los ejes ópticos de las cámaras son paralelos $\overrightarrow{O_R O_R} \parallel \overrightarrow{O_L O_L}$. Ambas cámaras tienen la misma distancia focal, f , con centros O_L y O_R separados una distancia B , llamada línea base (*baseline*), de forma que las imágenes que se forman, I_L e I_R , estén en planos paralelos. De esta manera la línea base es paralela a la coordenada \vec{x} de las imágenes. Con el modelo *pinhole* considerado, un punto en el espacio tridimensional P , con coordenadas homogéneas $(X, Y, Z, 1)^T$, se proyecta en cada una de las imágenes bidimensionales en los puntos p_L y p_R , con coordenadas homogéneas $(x_L, y_L, 1)^T$ y $(x_R, y_R, 1)^T$, respectivamente.

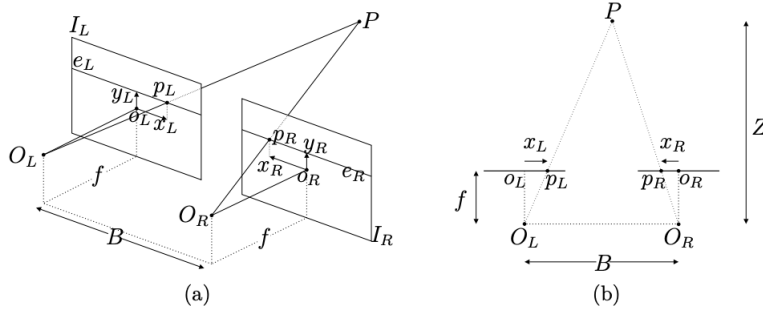


Figura 18: Profundidad a partir de la disparidad

El plano que contiene a los puntos P , O_L y O_R , interseca a las imágenes en dos rectas e_L y e_R . Dada la configuración del par estéreo, éstas son rectas epipolares entre sí, o sea, un punto, p_L , en la recta e_L de la imagen I_L tiene su correspondiente en algún punto de la recta e_R . Esto reduce la búsqueda del correspondiente de p_L de toda la imagen I_R a la recta e_R . En la Figura 18 (b) podemos ver cómo se relacionan los parámetros definidos en el par estéreo, que permiten obtener la relación entre la disparidad d y la profundidad Z del punto P . La disparidad es la diferencia en las coordenadas horizontales de los puntos p_L y p_R , o sea, $d = x_L - x_R$. Dependiendo del sistema de referencia utilizado en las imágenes, la definición puede cambiar de forma que el signo sea siempre positivo. Las coordenadas de p_L y p_R quedan relacionadas mediante $x_L = x_R + d$ y $y_L = y_R$. Por semejanza entre los triángulos $PO_L O_R$, $p_R O_R O_L$ y $p_L O_L O_R$ se llega a la relación entre d y Z :

$$Zd = fB \quad (8)$$

De esta forma se puede recuperar la profundidad de cada pixel en cada una de las imágenes a partir de la disparidad calculada [8].

Una vez explicado el significado de la disparidad, se tiene que parametrizar en el sensor mediante la ventana que se muestra en la Figura 19.

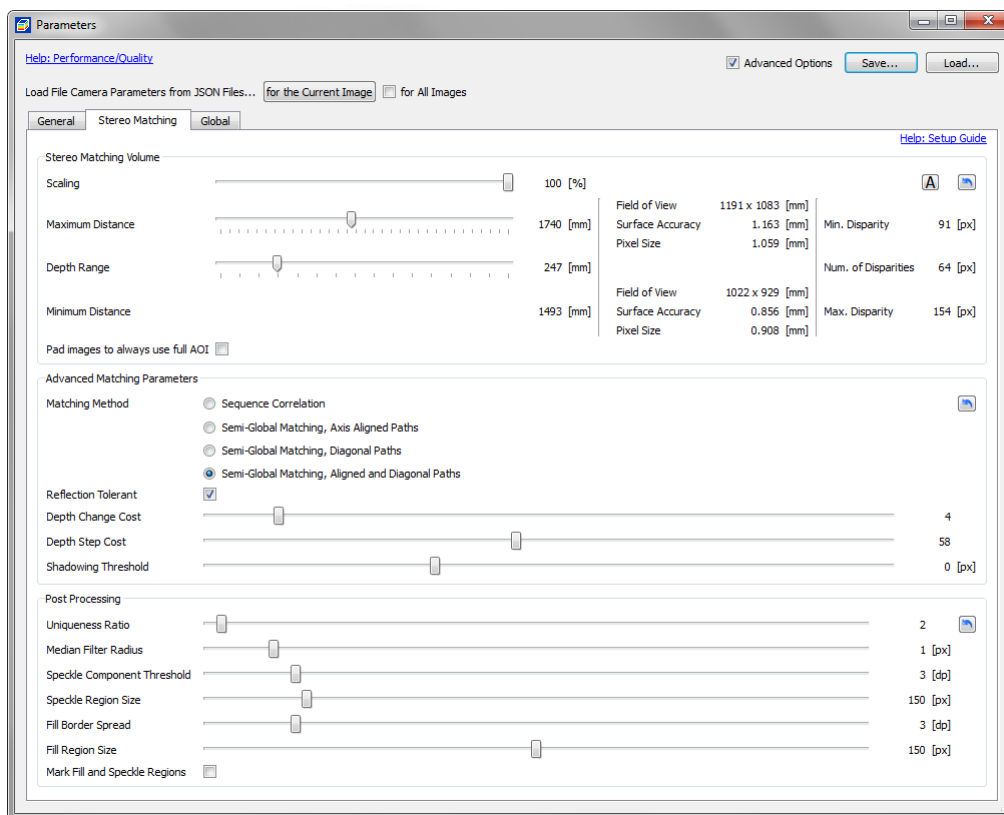


Figura 19: NxView: Ventana de parámetros

Dependiendo de la escena y la distancia a la misma, la construcción de la nube de puntos con los parámetros por defecto dará una nube incompleta y ruidosa. Para maximizar la correcta reconstrucción de profundidad se deberán ajustar los tres parámetros dispuestos en la parte superior de la ventana de parametrización:

- **Escalado:** disminuye la escala de la captura, obteniendo una nube con menor resolución en profundidad. Este parámetro debe estar al 100 % si se requiere del sensor una alta precisión.
- **Distancia máxima:** distancia máxima en la que el sensor procesará la disparidad. En este caso, esta distancia debe de ser siempre la distancia entre el sensor y la banda de transporte, 1100mm.
- **Rango de profundidad:** rango entre la distancia máxima y mínima a procesar por la disparidad. Este rango tiene que abarcar el área de trabajo del sensor con el fin de capturar hasta la pieza más alta, que en este caso se encuentra entre los 400mm. En este proyecto esta distancia se encuentra a 500mm de la banda de transporte, lo que hace un rango de trabajo de 600mm.

Una vez ajustados al mejor resultado, se continuará ajustando los parámetros más avanzados. Estos parámetros no actuarán sobre la cantidad de puntos que se obtiene, actuará sobre la forma en que se obtienen.

- **Disparidad** (maching method): este método especifica el tipo de optimización que se lleva a cabo en la función de mayor coste computacional de la disparidad. Esta optimización se lleva a cabo mediante los píxeles vecinos.
 - *"SgmAligned"*: sólomente se tiene en cuenta en el cálculo la 4-conectividad del píxel.
 - *"SgmDiagonal"*: sólomente se tiene en cuenta en el cálculo la 4-conectividad del píxel de forma diagonal.
 - *"SgmAlignedAndDiagonal"*: se tienen en cuenta todos los píxeles de alrededor del píxel a calcular, es decir, la 8-conectividad del mismo. Esta opción proporciona los mejores resultados pero también los más lentos.

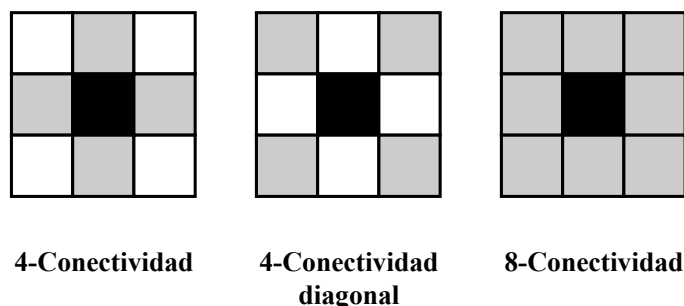


Figura 20: Tipos de conectividad

- **Tolerancias de reflexión:** la cámara no debe estar dispuesta de tal manera que se observen objetos con altos niveles de reflexión con ángulos cercanos a 90°. Esto es debido a que fuentes de luz dispuestas perpendicularmente al plano que observa el sensor provocan reflexiones especulares que son absorbidas por el mismo induciendo errores en el cálculo de la disparidad. Si se activa este parámetro, esos errores de reflexiones no se tendrán en cuenta en el cálculo de la disparidad.

Concretamente en este proyecto, los objetos a capturar son piezas de plástico con muy poco índice de reflexión y no se tendrán este tipo de problemas. No obstante, sí que han surgido problemas con lo referente al diseño de las placas de calibración y de masters (Sección 8) debido al material con el que se construyeron. Este material es el aluminio. Como el aluminio tiene un índice de reflexión muy alto, provocaba estos errores en la disparidad. Este problema se puede solucionar de varias maneras: sustituyendo el aluminio por otro material de menor índice especular, rotando la cámara para evitar las reflexiones directas o cubriendo el área de trabajo del sensor con una placa que evite las fuentes directas de luz. Se optó por la tercera solución.

- **Coste de variabilidad en profundidad:** Este parámetro controla error de los cambios de disparidad sobre píxeles sucesivos en las direcciones optimizadas. Un valor alto resultará en superficies menos abruptas pero algunos detalles de la captura se perderán.
- **Coste de paso en profundidad:** Este parámetro controla el error en las diferencias de entre pasos en la disparidad (por ejemplo, discontinuidades en la superficie) a lo largo de píxeles sucesivos en las direcciones optimizadas. Un valor alto mejorará la conectividad entre puntos de la superficie que dará lugar a menos agujeros en la captura. Pero un valor muy alto puede resultar en pérdidas de detalles y en bordes más bruscos.
- **Umbral de sombreado:** Como es de esperar, cuando se dispara el patrón de luz azul por el sensor, se crean sombras. Estas sombras serán más grandes conforme más altas sean las piezas. Como el sensor tiene dos cámaras, la sombra se captará de una forma u otra en ambas. Debido a la interpolación de subpíxeles y a los píxeles medio ocluidos, es razonable permitir pequeñas desviaciones de coincidencias entre la imagen de una cámara con respecto a la otra. Estas desviaciones se permitirán o no mediante este umbral.

Todos los parámetros anteriores han servido para optimizar lo mejor posible la algoritmia encargada de resolver la profundidad en los objetos capturados por el sensor.

Por último, existen otros parámetros de adquisición que manipulan toda la captura de forma que se consiga una mayor similitud con los objetos reales.

- **Relación de singularidad:** Filtrado de píxeles dependiendo de la singularidad de la correspondencia encontrada. Esta relación indica el porcentaje en el cual el coste de la siguiente mejor correspondencia debe de ser mayor para que ese pixel se acepte. Por ejemplo, siendo r la relación de singularidad y C_{1st} y C_{2st} el coste de la mejor y segunda mejor correspondencia en un pixel, el valor de profundidad de ese pixel solo será aceptado si

$$C_{1st} < \frac{100 - r}{100 \cdot C_{2st}} \quad (9)$$

Si se cumple la relación de singularidad $r = 10$, quiere decir que en cada píxel C_{1st} tiene que ser menor que $0,9 \cdot C_{2st}$. Si no es así, la profundidad de ese píxel se descartará y se establecerá como no válida.

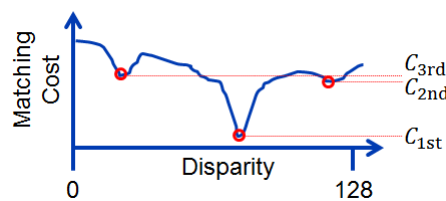


Figura 21: Relación de singularidad

- **Radio del filtro de mediana:** Este parámetro especifica el tamaño del filtro de mediana como el radio en píxeles, excluyendo el píxel central. Este filtro se aplica al mapa de disparidad y reduce el ruido en las figuras captadas manteniendo el borde definido. Si se define con valores altos, se puede llegar a perder la definición de la forma de la pieza, redondeando esquinas, etc. Por ello, este filtro debe de utilizarse lo menor posible.
- **Umbral de la componente de partículas:** define la división de la imagen en regiones para el filtrado de partículas o manchas. Siempre que dos disparidades de píxeles vecinos difieran más de lo impuesto en este parámetro, estos pertenecerán a regiones diferentes. En consecuencia, cada región resultante no tendrá discontinuidades mayores o iguales a este parámetro en su área del mapa de disparidad. Cuanto más pequeño sea este umbral, más pequeñas serán las regiones de disparidad resultantes.
- **Tamaño de la región de partículas:** Tamaño en píxeles de la región del mapa de disparidad debajo de la cual la región se eliminará del mapa de disparidad. El cálculo de estas regiones se realiza mediante el umbral de la componente de partículas visto en el punto anterior.
- **Umbral de extensión del borde:** Define qué regiones se llenarán estableciendo un umbral en la extensión máxima de las disparidades en el límite de la región o borde de la captura. Establecer este valor razonablemente pequeño asegurará que solo se llenen los bordes dentro de las caras planas, mientras que los espacios en las discontinuidades de profundidad se mantendrán sin llenar.
- **Tamaño de llenado de región:** Define el límite superior del tamaño de la región en píxeles, hasta el cual se acepta una región para ser llenada.
- **Marcar regiones llenadas y de partículas:** Este parámetro solo se utiliza en el caso de querer contrastar las regiones llenadas.

7 Tratamiento de la información

En este apartado del proyecto se detalla el procesamiento llevado a cabo en las nubes de puntos para que se pueda realizar la corrección de manera adecuada y controlada. Este tratamiento es debido a que el sensor capta todo lo que se encuentra en el campo de visión del mismo.

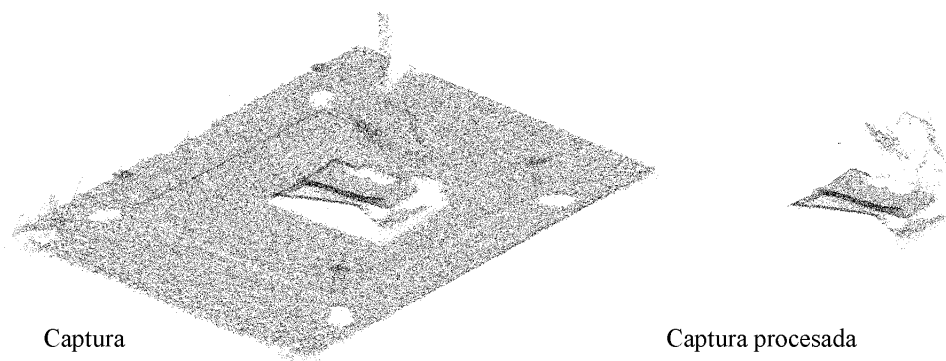


Figura 22: Comparativa del procesado de una nube de puntos

Si se observa la Figura 22, la captura sin procesar tiene muchos más puntos que la procesada debido a que en el campo de visión de la cámara se percibe el suelo donde se asenta la pieza a capturar. Si se decidiera realizar la ICP entre nubes de puntos sin procesar, el coste computacional sería muy elevado y el tiempo de procesado se vería aumentado. Además, conceptualmente, como se ha explicado en la Sección 4, el algoritmo funciona correctamente cuanto más parecidas sean las nubes de puntos. Si se interpone otro objeto que no esté presente en la captura de referencia, el algoritmo no dará buenos resultados. Por ello, se han utilizado filtros que se aseguren de que el algoritmo funcione con cierta seguridad. Filtros que son capaces de eliminar objetos tales como el suelo mediante un recorte por umbralización en profundidad. A continuación, se muestra un pequeño diagrama de bloques con los filtros diseñados en el programa LabVIEW.

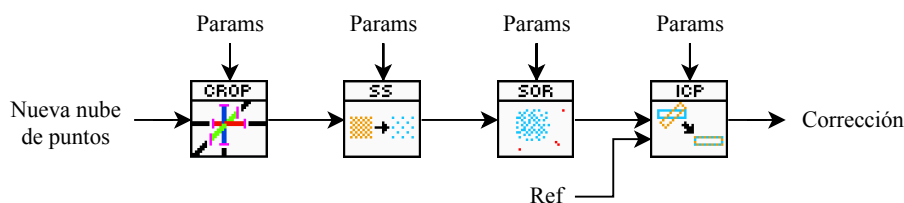


Figura 23: Filtros de procesado de las nubes de puntos

Crop) Primeramente se descarta todo lo que no pertenezca a la pieza recortando la nube de puntos con unos valores que permitan deshacernos de partes como el suelo o cualquier anomalía creada por el propio sensor por los bordes de la captura. Este filtro se coloca el primero debido a que es el que más puntos descarta, provocando así una mayor fluidez en los filtros posteriores. Cabe destacar que este filtro no utiliza la librería CCLib por su fácil implementación en el propio LabVIEW.

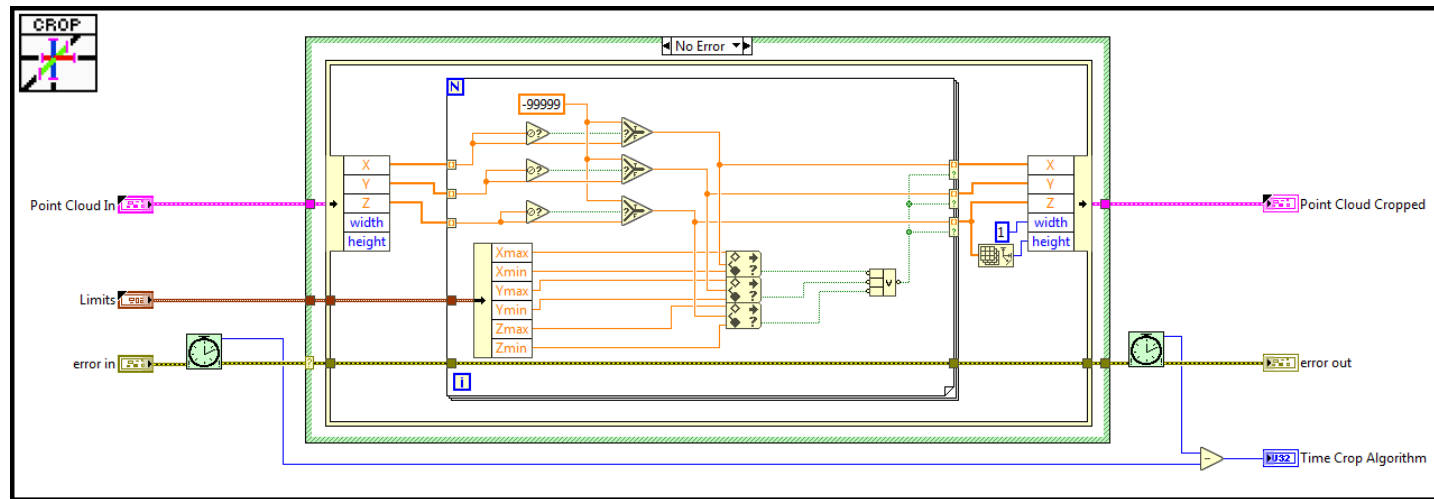


Figura 24: Interior del filtro de recorte LabVIEW

SS) Seguidamente, el filtro de remuestreo de la pieza (SubSampling) disminuye la densidad de puntos de toda la pieza a un número de puntos viable para que la comparación tenga suficiente información y al mismo tiempo aumentar lo menos posible el tiempo de procesamiento. Para este filtro es necesario utilizar la librería CCLib. A continuación, se muestra una parte del script de llamada a dicha librería a la función que corresponde al remuestreo.

```

1 //CCLib - SS
2 [...]
3 if (p_in->size() > remaining_points) {
4     SampledCloud = CCLib::CloudSamplingTools::subsampleCloudRandomly(p_in, remaining_points);
5 }
6 [...]

```

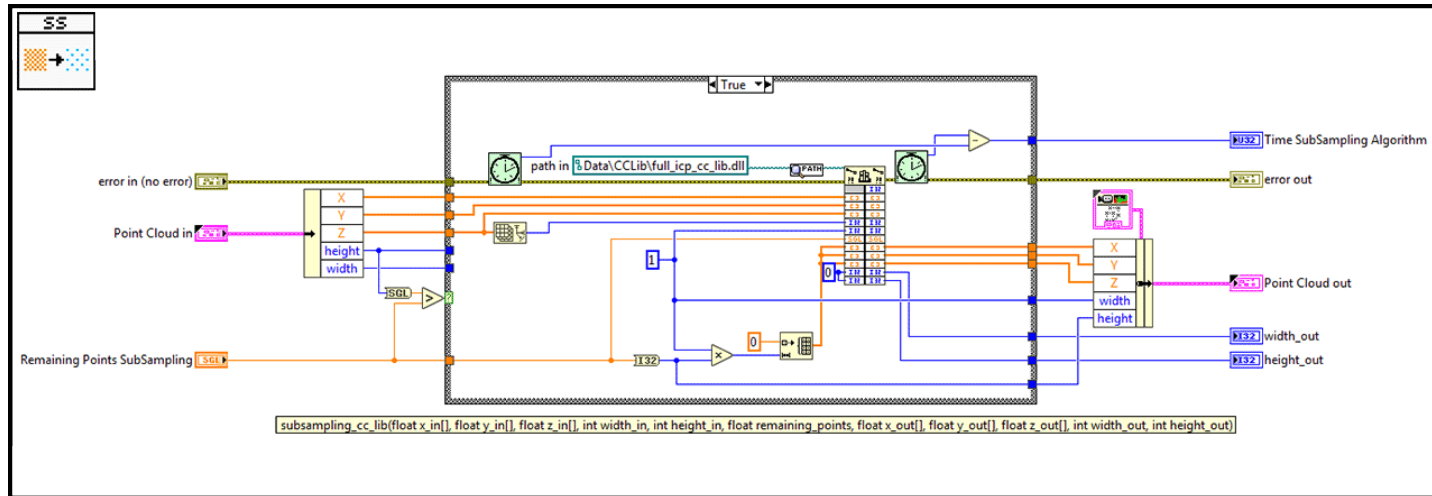


Figura 25: Interior del filtro de resampleo LabVIEW

37

SOR) Una vez remuestreada la captura, se procede a eliminar puntos dispersos que no pertenecen a la pieza, sino que son interferencias con el sensor o errores de cálculo del mapa de profundidad. Su funcionamiento se basa en la cercanía de los puntos vecinos. Por tanto, si se impone un umbral en la cantidad de vecinos que tiene un punto, se descartarán aquellos que no pertenezcan a zonas pertenecientes a la pieza en sí. CCLib también tiene una función para este filtro que se puede ver en el fragmento de código siguiente.

```

1 //CCLib - SOR
2 [...]
3 SampledCloud = CCLib::CloudSamplingTools::sorFilter(p_in, vecinos, desviacion);
4 [...]

```

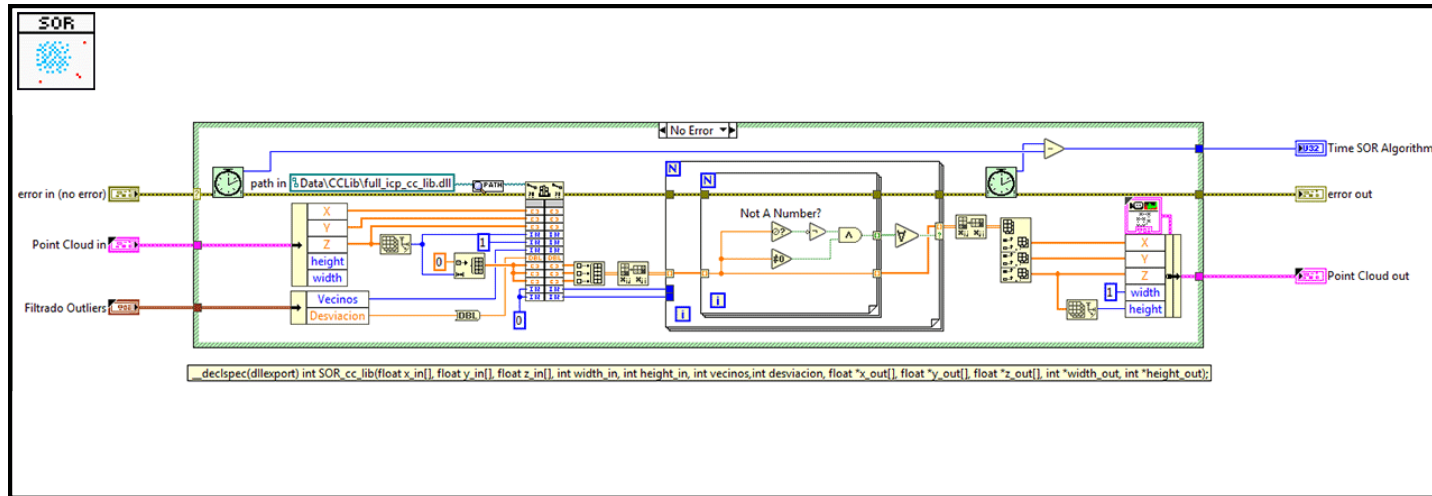



Figura 26: Interior del filtro de SOR LabVIEW

38

ICP) Por último, se realiza la ICP. Esta comparación se realiza con una nube de puntos procesada con los filtros explicados anteriormente y una nube de puntos referencia ya procesada con anterioridad. El resultado de dicha comparación es una matriz de transformación que, teniendo en cuenta el sistema de referencia en el que se encuentran las estaciones del proyecto, corregirá la rotación o desplazamiento que tenga la captura con respecto a la de referencia. Para realizar la comparación, se va a utilizar la librería CCLib de la siguiente forma.

```

1 //CCLib - ICP
2 [...]
3 params->transformationFilters = FilterTransformation;
4 params->samplingLimit = samplingLimit;
5 if (!convergenceType) {
6     params->convType = CONVERGENCE_TYPE::MAX_ITER_CONVERGENCE;
7 } else {
8     params->convType = CONVERGENCE_TYPE::MAX_ERROR_CONVERGENCE;
9 }
10 params->nbMaxIterations = maxIter;
11 params->minRMSDecrease = minRMSDecrease;

```

```

12 params->finalOverlapRatio = finalOverlapRatio;
13 params->filterOutFarthestPoints = filterOutFarthestPoints;
14 [...]
15 // Calcular ICP
16 CCLib::ICPRegistrationTools::Register(tgt, 0, src, *params, *transformation, RMS, pointcount);
17 [...]

```

39

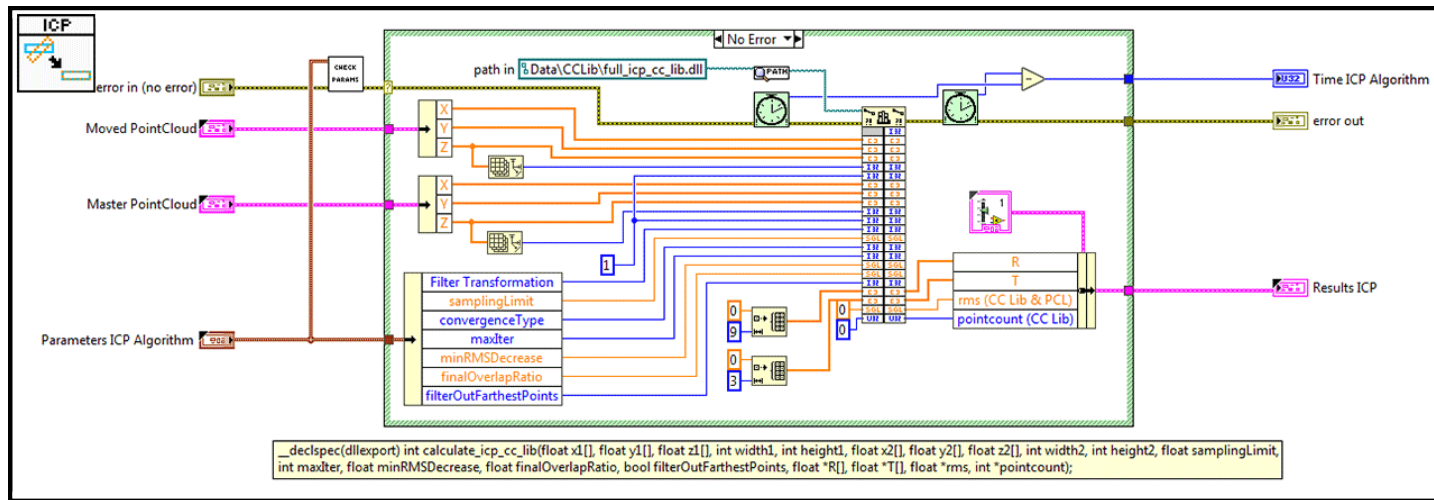


Figura 27: Interior de la ICP LabVIEW

7.1 Parametrización del procesado

Para conseguir el mejor resultado en el procesado de las muestras, se necesita comprender cada uno de los parámetros de entrada que puede tener cada filtro.

El primer filtro se trata del filtro de recorte *Crop* que tiene 7 parámetros de entrada. Estos parámetros son la nube de puntos a recortar y los límites inferiores y superiores en cada una de las dimensiones XYZ. Con esto se crea un cubo hipotético con el cual se desechará todo punto perteneciente a la muestra que se encuentre fuera del mismo. Con ello, se consigue eliminar grandes partes de la nube de puntos que no se requieren, como el suelo. Cabe destacar que si el sensor no está bien calibrado, este recorte se haría de forma irregular, puesto que los puntos de la muestra se encuentran en coordenadas irreales como, por ejemplo, que el suelo se encuentre de manera oblicua a la captura. Esto haría que parte del suelo entre en el cubo de recorte y se cuele en la captura junto a la pieza ocasionando un mal funcionamiento del algoritmo de correcciones.

El siguiente filtro es el de remuestreo *SS*. Hay varias formas de realizar un remuestreo de una nube de puntos según la librería CCLib: mediante el árbol octal, por espacio o de forma aleatoria [Figura 25]. Dependiendo de cada uno, se tienen unos parámetros de entrada u otros. Para este proyecto se han probado todos ellos, ofreciendo mejores resultados el remuestreado aleatorio. Esto tiene una explicación.

El remuestreo mediante árbol octal es capaz de averiguar la forma del objeto y remuestrear según dicha forma, pero no todas las muestras son perfectas. Se capturaban muestras que contenían agujeros en ellas, lo que hacía que la estructura del árbol octal cambiara, siendo el remuestreo diferente para cada una de las piezas.

El remuestreo por espacio elimina puntos cercanos entre si una distancia dada. En este caso, esa distancia, al ser la misma entre todos los puntos de la nube, crea un patrón que confunde al algoritmo de correcciones ICP. Debido a que la distancia entre puntos de una nube y la misma distancia entre puntos de la nube a corregir, el algoritmo converge rápido y a un resultado que es más afín a la similitud entre distancias de puntos que entre las nubes de puntos en sí.

Dados los problemas que han ido surgiendo con los dos métodos anteriores, si remuestreamos una nube de puntos de manera aleatoria, se evitan patrones de ningún tipo y la ICP tendrá en cuenta ahora si la totalidad de la pieza y la media entre distancias de forma controlada. Este remuestreo aleatorio tiene como parámetros la nube de puntos a tratar y el número de puntos de la nube a la salida del filtro.

Siguiendo el orden de procesado, ahora se encuentra el filtro de eliminación estadística de

valores atípicos SOR. Este filtro tiene 3 parámetros de entrada: la nube de puntos, los K-vecinos y la desviación. Los K-vecinos marcan el mínimo de puntos cercanos entre si para considerarse como un valor atípico y la desviación es el porcentaje de diferencia entre la distancia de un punto a otro y la media de las distancias entre los otros.

Y por último, se tiene el algoritmo ICP. Es importante que la nube de puntos llegue lo más preparada posible para que el cálculo de este algoritmo sea rápido y preciso. Los parámetros de la ICP se explican a continuación:

- **Límite de puntos:** Número de puntos límite a usar en la corrección. Cuantos más puntos se hallen en el cálculo, mayor será el tiempo de procesado pero mayor precisión en la corrección.
- **Criterio de convergencia:** El algoritmo es iterativo, y por lo tanto, tiene que converger en un punto. En este caso, se puede escoger entre converger vía número máximo de iteraciones o vía error mínimo entre iteraciones.
- **Número de iteraciones máximo:** Si se elige este criterio de convergencia, el algoritmo terminará de calcular al llegar al número de iteraciones marcado. Este tipo de convergencia asegura pasar un número exacto de iteraciones y te asegura también que el tiempo de procesado va a ser siempre el mismo pero no te asegura un error mínimo.
- **Número mínimo de RMS entre iteraciones:** Si se elige este otro criterio, el algoritmo convergerá llegado el punto en el que la distancia media cuadrática RMS entre iteraciones sea menor que el valor impuesto en este parámetro. Cuanto menor sea este número, mayor precisión se obtendrá en la corrección pero se tardará más tiempo en el procesado. Esta convergencia asegura el obtener una distancia media cuadrática o un error mínimo pero no te asegura cuánto vaya a tardar la corrección.
- **Índice de superposición:** Valor en tanto por ciento que muestra si la corrección debe de tener en cuenta toda la pieza o una parte de la misma con respecto a la otra. Es decir, si este valor se acerca a 0 significa que la pieza a corregir forma parte de un pequeño tanto por ciento de la pieza de referencia. Un ejemplo sería que la pieza a corregir fuera una nube de puntos de un manillar de una puerta y la pieza de referencia sea una puerta entera. Por el contrario, cuanto más se acerque a 1, la pieza a corregir y la pieza de referencia se tomarán como piezas que son idénticas y deben de converger todos sus puntos a una distancia mínima.
- **Filtro de puntos alejados:** Si este filtro se activa a 1, no se tendrán en cuenta puntos que no forman parte de la pieza. Este filtro es similar al filtro de eliminación estadística de valores atípicos.



- **Filtro de transformación:** Dependiendo del valor que tome este parámetro, las correcciones del algoritmo se darán en unas dimensiones específicas. Es decir, con este valor puedes decidir si las correcciones sólo van a tener en cuenta traslaciones en la dimensión X o rotaciones en el eje Z, etc.

8 Diseño de la estación de posicionamiento

El diseño de la estación debe de cumplir una serie de requisitos técnicos que se derivan de las condiciones de captura del propio sensor. Estos requisitos son:

- El apoyo del sensor debe de estar orientado hacia abajo y a una distancia de $600mm$ del inicio del área de trabajo.
- La estructura debe de permanecer completamente fija.
- La estructura debe de incorporar una sujeción para el cableado necesario para la alimentación y comunicación del sensor.
- La estructura no debe interferir en el área de trabajo del sensor.

Teniendo en cuenta estos requisitos técnicos, esta es la idea que se llevó a cabo por el departamento de ingeniería mecánica. Como se puede observar en la Figura 28, la estructura está compuesta mayormente por perfil estructural de acero y perfiles de aluminio. Se disponen dos columnas atornilladas al suelo alejadas del área de trabajo del sensor y los apoyos y atornillados necesarios para que el sensor quede totalmente anclado a la altura que corresponde.

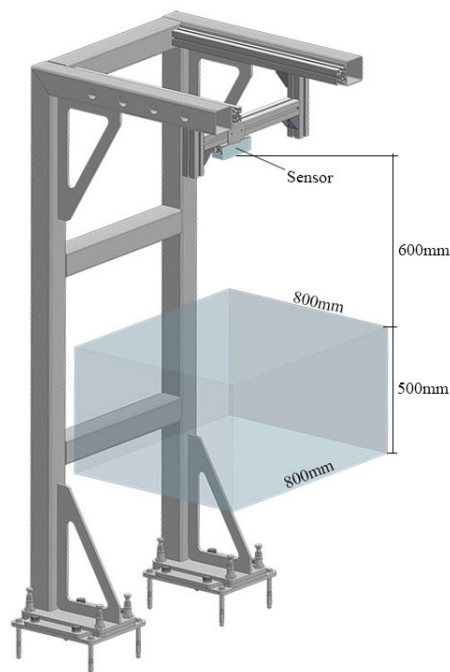


Figura 28: Diseño de la estación ST15

Además del diseño de la estación, se tienen que diseñar dos placas. Estas placas, al contrario que la estructura, se deben colocar dentro del área de trabajo del sensor. Más concretamente, encima de la cinta de transporte.

8.1 Placa de calibración

La placa de calibración debe de contener el patrón Halcon necesario para la calibración extrínseca del sensor y debe de ser visto completamente por el sensor.

Pero esta placa no solamente debe de tener esta calibración. Se tiene que diseñar teniendo en cuenta que el origen de coordenadas que se impone cuando calibras el sensor, también tiene que ser exactamente el mismo para los robots en las estaciones posteriores. Por ello, se tiene que añadir una terminación en forma de punta que será el el origen de coordenadas de todas las estaciones de inspección y del sensor. Sabido esto, el diseño de la placa se realizará teniendo en cuenta las siguientes características:

- El patrón debe de poder verse completamente por el sensor.
- La placa ha de poderse colocar en las estaciones *ST15*, *ST20* y *ST30*.
- Una vez colocada la placa, debe quedarse fija.
- En la placa se dispondrán de terminaciones para calibrar el origen de coordenadas del sensor y los robots.

Teniendo en cuenta estos requisitos, se obtuvo el siguiente resultado.

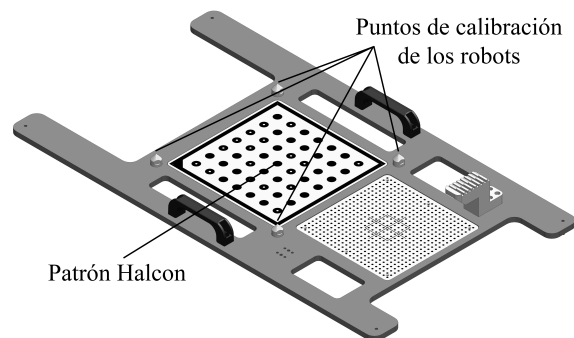


Figura 29: *Diseño de la placa de calibración*

8.2 Placa de referencias

Esta placa sirve como apoyo para las piezas de referencia o masters que mas adelante se necesitan para el algoritmo ICP. Estas masters son piezas de inyección sin ningún defecto que, mediante apoyos específicos para cada tipo de pieza, se encajan a la perfección en la placa. Como pasa con la placa de calibración, esta placa también se debe de poner en todas las estaciones de inspección sin que se desplace. Por lo tanto, el diseño de la placa se realizará teniendo en cuenta las siguientes características:

- Las piezas master deben de poder capturarse con el sensor.
- La placa ha de poderse colocar en las estaciones *ST15*, *ST20* y *ST30*.
- Una vez colocada la placa, debe de quedarse fija.
- Se deben construir soportes para todos los tipos de pieza que se inspeccionen en el proyecto.

Teniendo en cuenta los puntos anteriores, se obtuvo el siguiente resultado.

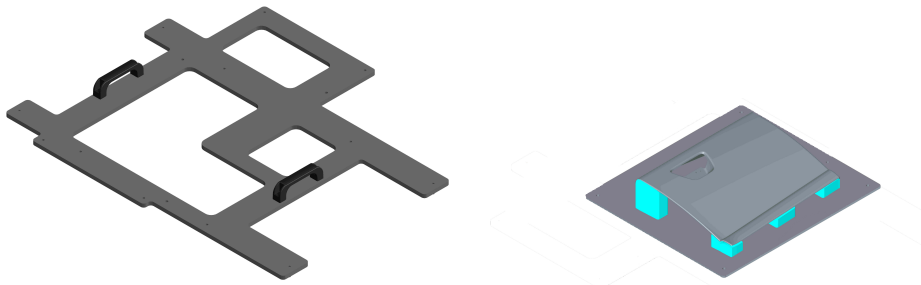


Figura 30: Diseño de la placa de referencias

9 Implantación del sistema

Éste es el apartado más costoso puesto que hay que llevar a la práctica lo estudiado durante el proyecto y su validez de su correcto funcionamiento. En un trabajo de esta magnitud primeramente se lleva a cabo la construcción del mismo en las instalaciones de la empresa realizadora. A esta fase se la conoce como *modular build*.

En cuanto se cumplan unos requisitos mínimos de funcionalidad pactados con los clientes, el proyecto se traslada a las instalaciones del cliente, donde se realizarán las últimas pruebas antes de validar la estación completamente. Esta es la última fase del proyecto y se la conoce como puesta en marcha (PEM).

9.1 Modular build

En esta fase, todos los departamentos deben realizar su trabajo siguiendo los planes acordados durante todo el proceso de investigación del proyecto.

Por parte de este trabajo, una vez se disponga de todas las herramientas diseñadas en la Sección 8, el sensor se colocará a $1100mm$ de la banda de transporte y de forma que quede anclado de manera paralela a la superficie de la misma. De esta forma, ya se pueden realizar los siguientes puntos para dejar la estación dispuesta para su explotación:

- Se calibrará.
- Se configurarán todos los parámetros de sistema de coordenadas.
- Se configurarán todos los parámetros de adquisición.
- Se obtendrán las referencias.
- Se calcularán en piezas nuevas las correcciones.

Por último, recalcar de esta fase que se debe comprobar la robustez y funcionalidad de la celda completa.

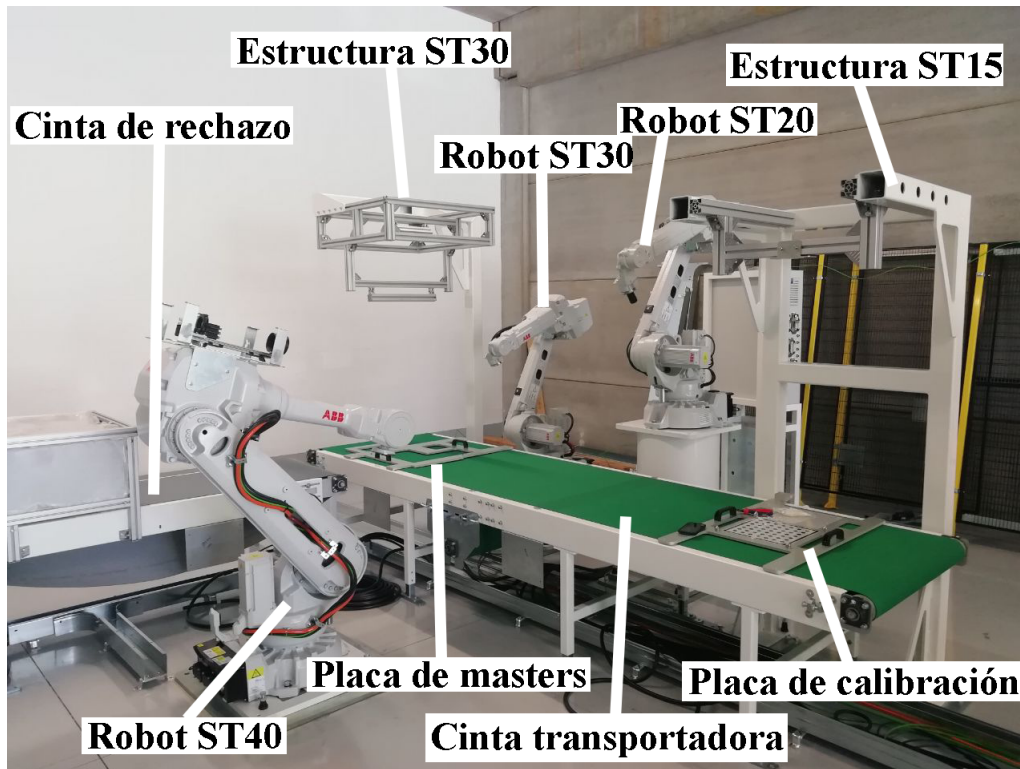


Figura 31: Modular Build

9.2 Puesta en marcha del sistema

La puesta en marcha es uno de los puntos más rigurosos de todo el proyecto. Esto es debido a que el proyecto, ya aprobado en el modular build, debe ahora implantarse en la fábrica del cliente. Ahora, debe de estar funcionando correctamente en la fábrica siempre. Es importante que se respete en la medida de lo posible todo lo establecido en el modular build. Se repetirán los pasos ya realizados en el modular build y se comprobará adicionalmente que el cambio de entorno de la instalación no afecta a su funcionalidad, como, por ejemplo, cambios en la iluminación que desarrollaron reflexiones en la captura del patrón de calibración que hizo que el sensor fuera incapaz de detectarlo. Esto implica modificaciones, como un techo que cubra el sensor, que pueden ser en sí mismas fuente de errores.



Figura 32: Estación ST15 en puesta en marcha

En la Figura 32 se puede observar la estación *ST15* instalada en la fábrica.

10 Evaluación de resultados

En este apartado se muestran los promedios de tiempo que se obtienen en la puesta en marcha del proyecto. Esto quiere decir que las piezas incluyen la variabilidad que se debe corregir para cumplir uno de los objetivos de este proyecto.

Modelo	Recorte [ms]	Remuestreo [ms]	SOR [ms]	ICP [ms]	Tiempo TOTAL [ms]	RMS Medio [adimensional]
Glove Box	30.667	3.970	79.030	7678.848	7792.515	2.018
Consola	30.923	2.865	110.673	5669.807	5814.269	1.750
Door Outer	31.444	4.555	90.148	9556.333	9682.481	1.590
Glove Box Open	32.905	6.151	99.692	3249.904	3386.250	1.641

Cuadro 7: Resultados

Si se observan los valores de tiempo de procesado total y se comparan con los tiempos de ciclo, se puede afirmar que son menores. Esto quiere decir que la estación cumple uno de los

objetivos del proyecto. Además, teniendo en cuenta que estas muestras se han tomado conforme a la variabilidad de las piezas en la dejada y el RMS, que indica la media de la distancia entre puntos, se encuentra entorno a 2 milímetros o inferior, también se puede decir que se cumple el otro objetivo propuesto para la estación.

11 Conclusiones

Estas investigaciones dieron lugar a la utilización de uno u otro método para conseguir el mejor resultado en la corrección de las piezas a inspeccionar y, como se ha podido ver en la evaluación de resultados, se puede concluir de este trabajo final de grado que se cumplen con los objetivos marcados por el proyecto.

En el proyecto se han llevado a cabo la investigación y su posterior implantación tanto del hardware como del software utilizados en la estación *ST15* que se encuentra dentro del proyecto de inspección de piezas de inyección de plástico. Investigaciones como el estudio conceptual de posicionamiento 3D, que implica la utilización de algoritmos iterativos para minimizar distancias entre puntos de una nube de puntos; la parametrización de la adquisición, que implica saber cuál es el funcionamiento del sensor de la estación de lo cual se extrae información interesante sobre el método de disparidad que realiza; y el procesado de las nubes de puntos, que es muy importante llevarlo a cabo para conseguir que la algoritmia funcione como es debido.

Se han encontrado diferentes escollos, al llevar a cabo de forma práctica los trabajos descritos que han sido superados siempre mediante métodos de depuración científicos, bien mediante la obtención y análisis de muestras bajo condiciones controladas o bien mediante análisis estadístico de resultados.

El proyecto ha requerido de la integración total en un equipo multidisciplinar, formado por ingenieros mecánicos, ingenieros de control y programación de robots, ingenieros de desarrollo de software e ingenieros eléctricos. Este hecho aporta una visión global, desde múltiples puntos de vista, con la que afrontar un proyecto complejo y completo en el campo de la ingeniería de la fabricación.

Como se destaca en la evaluación de resultados del presente documento, la estación desarrollada ha conseguido cumplir de manera eficaz todas las exigencias y objetivos planteados para la misma.

Referencias

- [1] LAVALLE, S., “*Planning Algorithms*”, Cambridge University Press, 2012.
<http://planning.cs.uiuc.edu/node102.html>
- [2] BESL, P. y MCKAY, N., “*A Method for Registration of 3-D Shapes*”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, nº 2, Febrero de 1992.
http://www-evasion.inrialpes.fr/people/Franck.Hetroy/Teaching/ProjetsImage/2007/Bib/besl_mckay-pami1992.pdf
- [3] ENSENSO[©] MANUAL, “*Manual*”, 2018
<https://www.ensenso.com/manual>
- [4] ENSENSO[©], “*Specification X36-5CP Series*”, 2018.
<https://www.ensenso.com/support/selector/model/?id=X36-5CP-8%2F16%2F1400-400%2F1600&minD=-227&numD=235&minX=3&maxX=2437&minY=212&maxY=1835>
- [5] RUSU, R., “*About PCL*”, Marzo de 2011.
<http://pointclouds.org/about>
- [6] GIRARDEAU-MONTAUT, D., “*3D point cloud and mesh processing software Open Source Project*”, Abril de 2014.
<http://www.cloudcompare.org/presentation.html>
- [7] SLOMP, M., KAWASAKI, H., DARIBO, I., SAGAWA, R., FURUKAWA, R., HIURA, S. y ASADA, N., “*Hardware-Accelerated Geometry Instancing for Surfel and Voxel Rendering of Scanned 4D Media*”, Kagoshima University, NII Japan, AIST Japan, Hiroshima City University.
<https://pdfs.semanticscholar.org/c686/3e9bf881dba1309f933f8b488d8b5edad1cb.pdf>
- [8] LECUMBERRY, F., “*Cálculo de disparidad en imágenes estéreo, una comparación*”, IIE - Fac. de Ingeniería, Universidad de la República, Uruguay, Octubre de 2005.
<https://iie.fing.edu.uy/publicaciones/2005/Lec05a/Lec05a.pdf>