Revista Iberoamericana de Automática e Informática Industrial

Revista Iberoamericana de Automática e Informática Industrial 19 (2022) 154-163

www.revista-riai.org

Calibración ojo a mano de un brazo robótico industrial con cámaras 3D de luz estructurada

Diaz-Cano, I.[a,*], Quintana, F.M.[b], Galindo, P.L.[b], Morgado-Estevez, A.[a]

[a]*Dpto. Ingeniería en Automática, Electrónica, Arquitectura y Redes, Escuela Superior de Ingeniería, Universidad de Cádiz - 11519, Puerto Real, Cádiz. España*
[b]*Dpto. de Informática e Ingeniería, Facultad de Ingeniería, Universidad de Cádiz-11519, Puerto Real, Cádiz, España*

To cite this article: Diaz-Cano, I., Quintana, F.M., Galindo, P.L., Morgado-Estevez, A. 2022. Eye-to-hand calibration of an industrial robotic arm with structured light 3D cameras. Revista Iberoamericana de Automática e Informática Industrial 19, 154-163. https://doi.org/10.4995/riai.2021.16054

Resumen

La visión artificial está cobrando cada día más importancia en el mundo de la robótica industrial, ya que es necesario realizar tareas cada vez más precisas y autónomas, por lo que se necesita un posicionamiento del robot más exacto. Para ello se precisa del apoyo de un sistema de visión que sea el que preste al robot precisión en su pose, calibrando dicho sistema con respecto al robot. Este trabajo presenta una metodología sencilla para abordar esta forma de calibración, llamada ojo a mano, empleando una cámara 3D de luz estructurada que obtiene la información del mundo real y un brazo robótico industrial de seis ejes. El método emplea el algoritmo RANSAC para la determinación de los planos, lo que supone una reducción notable de los errores, ya que las coordenadas de los puntos buscados, proceden de planos ajustados a miles de puntos. Esto permite que el sistema siempre va a tener la capacidad de obtener una matriz de transformación de las coordenadas de la cámara a la base del robot. Además el método propuesto se presenta ideal para realizar una comparación de precisión entre cámaras, debido a su sencillez y rapidez de uso. En este estudio se ha realizado el análisis de errores resultante utilizando dos cámaras 3D diferentes: una básica (Kinect 360) y otra industrial (Zivid ONE+ M).

Palabras clave: Calibración ojo a mano, Robótica industrial, Visión por computador aplicada a la robótica, Sistemas robóticos autónomos

Eye-to-hand calibration of an industrial robotic arm with structured light 3D cameras

Abstract

Computer vision is gaining more and more importance in the world of industrial robotics, since it is necessary to carry out increasingly precise and autonomous tasks, which is why a more exact positioning of the robot is needed. This requires the support of a vision system that is the one that gives the robot precision in its pose, calibrating said system with respect to the robot. This work presents a simple methodology to approach this form of calibration, called hand-eye, using a structured light 3D camera that obtains information from the real world and a six-axis industrial robotic arm. The method uses the RANSAC algorithm for the determination of the planes, which represents a notable reduction in errors, since the coordinates of the points sought come from planes adjusted to thousands of points. This allows the system to always have the ability to obtain a transformation matrix from the coordinates of the camera to the base of the robot. In addition, the proposed method is ideal for making a precision comparison between cameras, due to its simplicity and speed of use. In this study, the resulting error analysis was performed using two different 3D cameras: a basic one (Kinect 360) and an industrial one (Zivid ONE + M).

Keywords: Hand-eye calibration, Industrial robotics, Computer vision applied to robotics, Autonomous robotic systems

*Autor para correspondencia: ignacio.diaz@uca.es
Attribution-NonCommercial-NoDerivatives 4,0 International (CC BY-NC-ND 4,0)

1. Introducción

Hoy día la robótica en el ámbito industrial está tomando cada vez más relevancia gracias al desarrollo de los robots (Eriksson et al., 2008). La aplicación de robots industriales se está extendiendo rápidamente por todo tipo de factorías y empresas de todos los sectores y tamaños, sobre todo desde la aparición de la cuarta Revolución Industrial (Lu, 2017). No en vano se cuenta con la robótica autónoma como una de las tecnologías habilitadoras de este nuevo paradigma (Lasi et al., 2014). Para lograr esa autonomía en los robots la visión artificial, o visión por ordenador se ha posicionado como una de las tecnologías más usadas actualmente. Como comentan los autores en (Magnenat-Thalmann, 2020) son varios los campos donde la visión artificial es empleada: realidad virtual, aumentada y mixta, renderización y texturas, procesamiento de vídeos, modelados de terrenos, interacción humano-máquina, animación 2D, simulación visual, y la que nos ocupa, el procesamiento de imágenes.

Para lograr un procesamiento de imágenes preciso en el cual se pueda posteriormente situar un punto en el espacio es necesario contar con un dispositivo electrónico que capte la escena para luego ser procesada. En este escenario se presentan varias posibilidades: láser, láser 3D, lídar, cámaras de luz estructurada, tiempo de vuelo o estereoscópica. Una vez se tiene la posición desde las coordenadas del sistema de visión se pretende trasladar estas coordenadas al sistema de coordenadas del robot para que este finalice el trabajo en el objeto. Este tipo de calibración se conoce como ojo a mano (*hand-eye* en inglés), como se comenta de forma general en (Ali et al., 2019).

La calibración ojo a mano tiene como objetivo unificar el sistema de coordenadas del sistema de visión y el robot, de modo que la postura del objeto, determinada por el sistema de visión, se pueda convertir al sistema de coordenadas del robot. Así se obtiene un método de transformación de coordenadas que puede ser empleado en cualquier trabajo o posición que se lleve a cabo.

Los sistemas ojo a mano que más se emplean incluyen dos variantes *Eye-to-Hand* y *Eye-in-Hand*. En el primero de ellos, tanto la posición de la cámara como del robot son fijas, por lo que la relación ojo a mano se emplea para resolver la conversión de coordenadas entre el dispositivo de visión y la base del robot (Cui et al., 2020). Por otro lado, en el sistema *Eye-in-Hand*, el dispositivo de visión está fijado en el extremo del brazo robótico o efector final, resolviendo la relación ojo a mano mediante la conversión de coordenadas del sistema de visión al efector final (Liu et al., 2021).

En este artículo se presenta un estudio inicial donde se aplica una metodología sencilla para conseguir una calibración ojo-mano, con la variante *Eye-to-Hand*, empleando una única cámara de luz estructurada, de manera que el extremo del robot esté libre para albergar otra herramienta necesaria para realizar una tarea concreta: una pinza, un medidor, una antorcha de soldadura,... Además se aplican dos algoritmos conocidos, uno para la detección de planos, y otro para el alineamiento de la nube de punto, así como una técnica conocida, la descomposición en vectores singulares. Esta técnica que suele aplicarse para la solución de diversos problemas pero no es muy común situarla en una calibración de este tipo, aunque sí en otras calibraciones, como se explica en (Toan and Khoi, 2018). Para el proceso de calibración se ha construido una pieza de calibración específica. El método se ha aplicado a una cámara básica y a otra más avanzada y precisa, para comprobar la bondad del estudio presentado. Se destaca el empleo del algoritmo RANSAC para la detección de puntos clave que supone conseguir la intersección de planos a partir de miles de puntos estimados, permitiendo que el cálculo final se consiga en un solo paso *single-step*, sin aplicar algún algoritmo iterativo.

La estructura de este documento es la siguiente. En la Sección 2 se presentan una serie de trabajos relacionados con el estudio propuesto, y que resuelven la calibración ojo a mano empleando diferentes técnicas y dispositivos. A continuación, en la Sección 3 se explica el proceso de calibración seguido en este estudio. En la Sección 4 se muestran los resultados y discusión del trabajo. Por último, en la Sección 5 recoge las conclusiones del artículo y se propone algún trabajo futuro para mejorar y avanzar sobre la investigación presentada.

2. Trabajos relacionados

A través de esta sección se van a dar a conocer otros medios, trabajos y estudios que existen en la literatura que se encargan de resolver la calibración que nos ocupa en este trabajo, la calibración ojo a mano.

El problema de la calibración mano-ojo apareció por primera vez y recibió su nombre de la comunidad robótica, donde se montó una cámara, a la que se le denominó ”ojo“ en el efector final del robot, su ”mano“. Las cámaras se deben calibrar empleando un patrón. A partir de aquí, la transformación desconocida del sistema de coordenadas del robot al sistema de coordenadas del patrón de calibración, así como la transformación de la cámara al sistema de coordenadas de la mano, se deben estimar simultáneamente (Munchen, 2009).

Hay que tener en cuenta que cuando el robot se encuentra en distintas poses se consigue la relación entre el ”ojo“ y el objeto de calibración. Así se puede establecer una ecuación de calibración $AX = XB$. Esta ecuación fue resuelta por los autores en (Shiu and Ahmad, 1989), obteniendo la matriz de relación ojo a mano. El hecho de resolver esta matriz ya suponía obtener un pequeño error y una solución eficiente. En cuanto a la resolución final del problema, existen muchos métodos diferentes de calibración ojo-mano. Hay dos métodos principales para resolver la matriz de intercambio ojo-mano.

Por su parte los autores en (Tsai and Lenz, 1989) llegan a descomponer el problema en dos partes teniendo por un lado la matriz de rotación y por otro la matriz de traslación. En primer lugar se obtiene la matriz de rotación y a continuación se sustituye la matriz de rotación en el vector de traducción de la solución empleando una suma considerable de operaciones aritméticas.

En el estudio presentado en (Dias et al., 1991) se emplean dos cámaras de luz estructurada montadas en un robot de seis ejes de manera que consiguen la calibración ojo a mano mediante la creación de una matriz de calibración invariante al movimiento a la que se le aplican procesos de optimización para obtener una mejor aproximación en el error. Además se ayudan para ello de un filtro Kalman.

En la investigación presentada en (Zhao and Liu, 2009), los autores emplean la teoría del movimiento del tornillo, técnica algebraica empleada en otras soluciones robóticas (Featherstone, 2007). Con la aplicación de esta teoría se establece una ecuación matricial mano-ojo usando cuaternión, y obteniendo un resultado simultáneo de rotación y traslación mediante ecuaciones lineales. a partir de esto, el algoritmo que se propone tiene una alta precisión y eficacia computacional.

En (Hu and Chang, 2013) se aborda el problema de la calibración ojo a mano para la configuración *Eye-to-Hand*. Para ello, la ecuación de calibración clásica $AX = XB$ se ha obtenido descomponiendo por un lado la ecuación de rotación y por otro la de traslación. La matriz de rotación se presenta por el método de cuaternión unitario. Además, se tiene en cuenta la influencia negativa del ruido durante la calibración y propone una rectificación para mejorar la precisión de la calibración.

Los autores de (Lundberg et al., 2014) abordan el problema de la calibración de los valores intrínsicos de la cámara (aquellos que proporciona el hardaware y que pueden ser modificados en cada captura) junto a la calibración ojo a mano, empleando un sistema de visión junto a un marcador de un solo punto. En este caso el objeto calibrador es pequeño por lo que la recalibración y la verificación de la precisión es más rápida, sin alterar en demasía el entorno del robot. La solución propuesta proporciona una rutina de calibración que produce resultados de alta calidad sin necesidad de intervención manual.

En el área de la cirugía cada vez es más común emplear robots que necesiten estar bien posicionados, con una precisión exhaustiva, con el fin de poder realizar correctamente su trabajo. En el caso de (Zhang et al., 2017), sus autores emplearon el cuaternión dual con el que se encargaron de representar la transformación rígida. Posteriormente propusieron un método iterativo de dos pasos para recuperar la parte real y la parte dual del cuaternión dual simultáneamente, lo que podría aplicarse para estimar la rotación y traslación de la transformada rígida. De esta manera consiguieron una eficiente aproximación de posición entre el laparoscopio y el robot manipulador. El estudio propuesto por (Pachtrachai et al., 2018) se basa en la transformación adjunta de movimientos de torsión que resuelve el problema de forma iterativa mediante estimaciones alternas de rotación y traslación. Así, demuestran que este enfoque converge en una solución con una mayor precisión que las inicializaciones de forma cerrada.

Los autores en (Taryudi and Wang, 2018) proponen un sistema de calibración con una cámara estereoscópica donde, primero, se calibra una cámara estéreo para obtener los parámetros intrínsecos y extrínsecos de la cámara. Luego, las características de color específicas del objeto deseado se extraen utilizando algoritmos de procesamiento de imágenes. Por ultimo, según el centroide del objeto deseado, se calcula la posición 3D (3 dimensiones) del objeto en cuestión.

En el caso del estudio realizado por (Li et al., 2018) se propone un método de calibración ojo a mano y otro método robot-mundo en el que consiguen la precisión sin emplear una pieza de calibración. En este caso establecen un modelo matemático para realizar la transformación de la pinza del robot a la cámara, y transformación de robot-mundo utilizando el ajuste matemático llamado producto Kronecker (Uhlig, 1992).

Orientado a la soldadura robótica automatizada mediante arco eléctrico, los autores proponen en (Zou and Chen, 2018) un sistema de calibración ojo a mano en el que se ayudan de un láser de alta precisión. Además emplean el mínimo cuadrado ordinario (MCO) para calcular la matriz de rotación y el (*Semidefinite Programming*(SDP) (de Klerk, 2002) se utiliza para identificar los vectores de dirección de la matriz de rotación para asegurar su ortogonalidad.

A diferencia de los métodos tradicionales de calibración ojo a mano, el método propuesto en (Koide and Menegatti, 2019) se basa en la minimización de errores de reproyección. Para ello, toma imágenes directamente del patrón de calibración sin estimar explícitamente la pose de la cámara para cada imagen de entrada.

Con todo, y tras haber realizado un repaso en la literatura, el método propuesto se diferencia con respecto a los estudios hasta la fecha, en que es capaz de conseguir una calibración del sistema en un solo paso, habiendo automatizado gran parte del proceso. Además se han minimizado los dispositivos a emplear, ya que solo necesita una pieza de calibración, además de la cámara. Además el método presentado es flexible, es decir, se le pueden incorporar mejoras que lo hagan más robusto. Se habla en términos de robustez en el sentido en el que la aplicación del método, siempre es capaz de devolver una calibración ojo-mano del sistema.

3. Proceso de calibración

A lo largo de esta sección se describe la metodología seguida en este trabajo para lograr la calibración ojo a mano de una cámara de luz estructurada. La metodología se ha seguido empleando la tecnología de captura de imágenes por medio de luz estructurada. En ella se han minimizado los pasos que se deben dar hasta obtener la calibración, y se han automatizado los pasos a seguir. En la Figura 1 se muestran los pasos que se deben seguir. A continuación se explican detalladamente cada uno de los pasos del método propuesto.

3.1. Captura de puntos de calibración

Es necesario tener una pieza de calibración donde tomar una serie de puntos que sirvan para realizar la calibración con respecto al robot, se recomienda mínimo nueve puntos, aunque en nuestro caso se ha creado una pieza de calibración con quince puntos. Estos puntos serán tomados con el "end-effector". En el caso de tener una herramienta de trabajo, que será lo más habitual, los puntos se tomarán con dicha herramienta. Una vez tomados los puntos se guardarán en un fichero de texto. La herramienta del robot deberá estar calibrada previamente a tomar los puntos. Estos puntos serán considerados como las medidas reales sobre los que ajustaremos los puntos tomados con la cámara. La pieza de calibración se ha creado para este trabajo y se muestra en la Figura 2 con el orden en el que se tienen que tomar dichos puntos.

Figura 1: Pasos del proceso de calibración propuesto

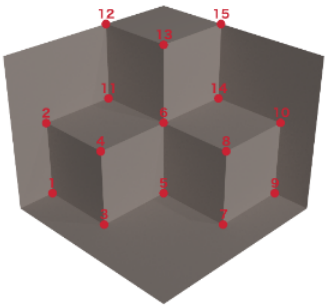


Figura 2: Pieza donde se recogen quince puntos en orden para el proceso de calibración cámara-robot.

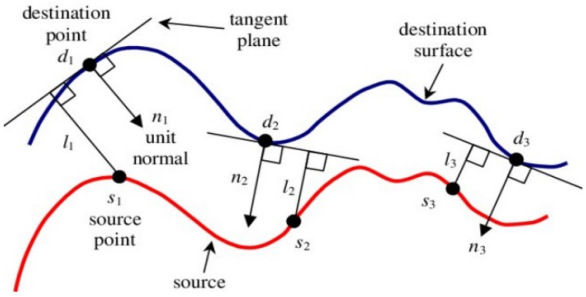


Figura 3: Error punto-plano. Cálculo de la distancia entre cada punto de la nube de puntos con la tangente del punto de destino correspondiente. Fuente: (Low, 2004)

3.2. *Captura del espacio 3D en coordenadas de la cámara*

En esta fase es necesario hacer una captura 3D con la cámara de luz estructurada de la pieza de calibración. La metodología se va a enfocar en el empleo de dos cámaras de luz estructurada que van a seguir los mismos pasos, cada una con las particularidades que ofrece. Así, en el caso de la cámara Kinect 360 se va a emplear el algoritmo KinectFusion creado por Microsoft Research en 2011 y que se explica en (Newcombe et al., 2011; Izadi et al., 2011). Este algoritmo necesita como entrada una secuencia temporal de mapas de profundidad producidas por la cámara Kinect 360. Al ser en tiempo real, se realiza una representación de la superficie extraída por cada *frame*, se alinean y unen con un modelo global. Así, se puede definir este algoritmo en los siguientes apartados:

- **Extracción de superficie**: En primer lugar se va a obtener de entrada un nuevo mapa de profundidad proporcionado por la cámara. Para que el algoritmo pueda funcionar, este mapa de profundidad debe de convertirse a una nube de puntos. Para realizar esta conversión la cámara guarda internamente una matriz de calibración donde la profundidad de los píxeles los convierte a coordenadas 3D. La normal de estos puntos se estiman mediante el producto vectorial del punto escogido junto con el inmediatamente superior y el inmediatamente a la derecha. Como resultado obtenemos una nube de puntos, con los datos de los vértices y las normales de cada uno.

- **Alineamiento**: En la primera iteración del algoritmo, la nube de puntos obtenida en el paso anterior corresponde al modelo global. En iteraciones posteriores, las nuevas nubes de puntos se alinean con el modelo y se unen. Este alineamiento se realiza mediante el algoritmo ICP(*Iterative Closest Point*), o punto más cercano iterativo. Dicho algoritmo y algunas de sus variantes son mostradas por los autores en (Rusinkiewicz and Levoy, 2001). Con el cuál también se estima la nueva pose de la cámara. El algoritmo ICP calcula de forma iterativa la transformación necesaria para minimizar el error cuadrático medio utilizado. Los errores más comunes son punto-punto y punto-plano. El primero asocia cada punto de la primera nube de puntos con el punto más cercano a la segunda nube de puntos y calcula su distancia. El error punto-plano, como se puede observar en la Figura 3, calcula la distancia entre cada punto de la primera nube de puntos con la tangente del punto de destino (Low, 2004).

- **Reconstrucción**: Una vez alineadas las nubes de puntos y estimada la nueva pose de la cámara, mediante el uso de una función TSDF (Truncated Signed Distance Function) se extrae sobre la nueva nube de puntos la superficie de los objetos en la escena y se asignan números negativos a aquellos píxeles que se encuentren dentro de los objetos o en un área que no se haya medido aún. Por otro lado, se asignan números positivos a los píxeles que se encuentran fuera de la superficie y 0 a los píxeles que se encuentran en la superficie, tal y como los autores explican en (Werner et al., 2014). Finalmente esto se une con el modelo global mediante una media ponderada calculando así la nueva superficie.

  En el caso de la cámara Zivid, por sus características es capaz de tomar una imagen de la pieza de calibración entera, sin necesidad de barrido, ni de aplicar ningún tipo de fusión de imágenes, mostrando una nube de puntos similar a la que ofrece el algoritmo KinectFusión. Una vez concluida la aplicación del algoritmo KinectFusión o la captura con la cámara industrial, se obtiene una nube de puntos como se muestra en la Figura 4.

3.3. *Estimación de vértices con el algoritmo RANSAC*

A continuación, a partir de la nube de puntos obtenida en el anterior paso, con una cámara u otra, se va a aplicar el algoritmo RANSAC (Fischler and Bolles, 1981) para buscar los puntos de calibración en el modelo escaneado, es decir, en la pieza de calibración.

Se trata de un algoritmo iterativo no determinista de ajuste de parámetros de modelos matemáticos a partir de un conjunto de datos que contienen multitud de valores atípicos. Este algoritmo es utilizado ampliamente en visión por computador y produce un resultado razonablemente eficiente con una cierta probabilidad como los autores también expusieron en (Yang and Förstner, 2010). Cada iteración del algoritmo se puede dividir en dos etapas:

- La primera consiste en seleccionar un subconjunto mínimo de datos seleccionados aleatoriamente para ajustar un modelo.

- En la segunda etapa, el algoritmo comprueba qué datos pertenecen al modelo estimado mediante un umbral de error. Los datos que sobrepasen ese umbral se consideran atípicos del modelo.

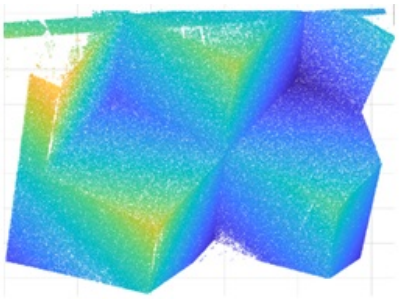


Figura 4: Nube de puntos resultante tras la captura en 3D de la pieza de calibración

Una vez ajustado un modelo y obtenido valores atípicos y valores que pertenecen al modelo, se calcula el número máximo de iteraciones que debe realizar el algoritmo. Inicialmente el número de iteraciones N es un valor alto para asegurar con una probabilidad p que determina si el algoritmo selecciona únicamente *inliers*(valores que pertenecen al modelo) en los n puntos del modelo estimado (Derpanis, 2010). Sea u la probabilidad de que cualquier punto seleccionado sea parte de un modelo y sea $v = 1 - u$ la probabilidad de observar un valor atípico. Entonces siendo n el número de puntos que son necesarios para seleccionar un modelo, u^n es la probabilidad de que todos estos puntos sean *inliers* y por consiguiente, que $1 - u^n$ sea la probabilidad de que al menos un punto sea un valor atípico. Esta probabilidad elevada al número de iteraciones que debe hacer N, es la probabilidad de que el algoritmo nunca seleccione un conjunto de n puntos donde todos estos son *inliers*, como se puede comprobar en (1).

$$1 - p = (1 - u^n)^N \tag{1}$$

En este estudio se va a emplear el algoritmo para ajustar planos a una nube de puntos, por lo tanto los datos que se manejan serán coordenadas 3D. El modelo propuesto necesitará 3 datos para ser estimado, es decir, el parámetro $n = 3$ quedando $1 - p = (1 - u^3)^N$. Ahora se pueden aplicar logaritmos para despejar el número de iteraciones, obteniendo (2).

$$N = \frac{\log(1 - p)}{\log(1 - u^3)} \tag{2}$$

Al aplicar el algoritmo a la pieza de calibración se obtienen nueve planos diferentes. Sabiendo esto se realiza la intersección de los mismos para encontrar los puntos clave del modelo, o puntos de calibración, quince en total.

3.4. Cálculo de la transformación geométrica

Una vez obtenidos los puntos de calibración el siguiente paso consiste en calcular la transformación geométrica entre ambos conjuntos de puntos (Los obtenidos de la nube de puntos y los reales). Para encontrar esta transformación se aplicará la descomposición en valores singulares de la matriz de covarianza entre los puntos de ambos conjuntos (SVD), que descompondrá una matriz M en 3 submatrices, $A = U\Sigma V^t$, donde $m > n$. A es una matriz mxn. U es una matriz ortogonal de dimensiones mxn. Σ es una matriz diagonal de dimensiones mxm y V es una matriz ortogonal de dimensiones nxn. Al ser las matrices U y V ambas ortogonales se cumple que $UU^t = U^tU = I$, por lo que al descomponer A en $A = U\Sigma V^t$, se cumple que:

$$\begin{aligned} A^tA = V\Sigma U^tU\Sigma V^t = V\Sigma^2V^t \\ (A^tA)V = V\Sigma^2 \end{aligned} \tag{3}$$

Por lo tanto se cumple que los vectores singulares derechos de A son los vectores propios de A^TA y los valores singulares la raíz cuadrada de los valores propios. Para encontrar la matriz U, el proceso sería análogo al de V pero con la matriz AA^T en lugar de A^TA. Pero como A^TA es del mismo tamaño o inferior que AA^T se puede encontrar U utilizando V y Σ que los calculamos anteriormente. Para ello despejando la ecuación inicial de la descomposición resulta que:

$$U = AV\Sigma^{-1} \tag{4}$$

La descomposición en valores singulares tiene múltiples aplicaciones: compresión de imágenes, cálculo de la pseudoinversa, ajuste de ecuaciones lineales, etc. Esto permitirá ajustar un plano a un conjunto de puntos y, como los autores comentan en (Sorkine-Hornung and Rabinovich, 2017), obtener la transformación entre distintas nubes de puntos mediante los vectores singulares izquierdos y derechos U y V. El objetivo de esta transformación es encontrar una matriz R de rotación y un vector t de traslación entre ambas nubes de puntos. Como resultado, para calcular la transformación entre dos conjuntos de puntos se debe obtener en primer lugar los centroides $\overline{p}$ y $\overline{q}$ de ambos conjuntos.

Una vez calculados los centroides se deben centrar ambos conjuntos con el origen de coordenadas restando los centroides de los puntos de origen. De forma general se aplicará la siguiente fórmula:

$$x_i = p_i - \overline{p}, y_i = q_i - \overline{q}, i = 1, 2, 3...n \tag{5}$$

A continuación, se calcula la matriz de covarianza entre los puntos centrados, donde X e Y son matrices de dimensiones dxn (En el caso en el que se va a aplicar esta teoría al ser puntos tridimensionales serán de $3xn$) que tienen a x_i e y_i como sus columnas respectivamente:

$$S = XY^t \tag{6}$$

Seguidamente se calcularía la matriz de rotación mediante la descomposición en valores singulares de $S = U\Sigma V^t$. Por último, se calcularía la traslación entre ambas nubes, como se muestra en (7).

$$t = \overline{q} - R\overline{p} \tag{7}$$

Así, aplicando la teoría expuesta anteriormente se va a calcular la transformación geométrica entre ambos conjuntos de puntos (Los obtenidos de la nube de puntos y los reales). Para ello se va a buscar una rotación y traslación eficiente que minimice el error cuadrático medio entre los puntos correspondientes, como se muestra en (8).

$$B = RA + t \tag{8}$$

R y t son la matriz de rotación y el vector de traslación respectivamente necesarios para transformar las coordenadas de los puntos A a los puntos B, es decir, de los puntos obtenidos en el modelo 3D a los puntos reales. Así, lo primero que nos indica la teoría es que se deben obtener los centroides. En (9) se puede ver cómo se ha aplicado en este estudio, obteniendo

$$P = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$centroideA = \frac{1}{N}\sum_{i=1}^{N} P_A^i \tag{9}$$

$$centroideB = \frac{1}{N}\sum_{i=1}^{N} P_B^i$$

Una vez calculados los centroides, siguiendo la teoría, se debe crear la matriz R que contendrá los puntos centrados en el origen. En el caso estudiado obtenemos

$$R = \sum_{i=1}^{N} (P_A^i - centroideA) - (P_B^i - centroideB)^t \tag{10}$$

A continuación se va a buscar una rotación eficiente de la matriz R utilizando la descomposición en valores singulares (SVD). Así, se descompondrá dicha matriz en 3 submatrices tales que:

$$\begin{aligned} [U, S, V] &= svd(R) \\ R &= USV^t \end{aligned} \tag{11}$$

El siguiente paso, según (6), a partir de la matriz de rotación se puede obtener la matriz de covarianza de los puntos centrados. En el caso estudiado se obtiene la siguiente ecuación

$$R = VU^t \tag{12}$$

Por último, se procede a crear la matriz de transformación en coordenadas homogéneas M, tendrá un formato como el que se muestra en (13).

$$P = \begin{pmatrix} R_{11} & R_{12} & R_{13} & t_1 \\ R_{21} & R_{22} & R_{23} & t_2 \\ R_{31} & R_{32} & R_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{13}$$

Para finalizar el proceso de calibración de este estudio esta matriz M se guardará en un fichero de texto, que se usará para convertir del sistema de coordenadas de la cámara al sistema de coordenadas del brazo robótico cada vez que este necesite hacer un trabajo de posicionamiento mediante la visión de la cámara.

Figura 5: Robot Fanuc LR Mate 200 iD, con la cámara Kinect 360 instalada en su parte superior, y una antorcha de soldadura en el efector final

4. Resultados experimentales

Con el fin de validar la idoneidad de la metodología propuesta para realizar una calibración ojo a mano de un brazo robótico con cámaras 3D de luz estructurada se ha realizado un estudio del error. A continuación se describen el entorno experimental y los resultados obtenidos.

4.1. Entorno de experimentación

Para realizar el experimento, la cámara 3D debe ir instalada en una posición que no sea el efector final del robot, quedando éste libre para poder colocar otra herramienta como una pinza, tenaza, martillo, taladro,...En nuestro caso se ha colocado una antorcha de soldadura, como se muestra en la Figura 5. Se ha empleado el brazo robótico Fanuc LR Mate 200 iD de seis grados de libertad. Será la herramienta instalada la que recogerá los puntos considerados como reales, de forma manual, como se indica en el método propuesto, estando dicha herramienta previamente calibrada mediante el método que recomienda el fabricante. La cámara 3D, primero la básica y posteriormente la industrial, se han instalado en el antebrazo del robot. Para ello se ha creado una pieza en impresión 3D (una por cada cámara) con la que sujetar firmemente la base de la cámara al robot y que no se mueva a la hora de capturar la escena.

Las cámaras con las que se ha contado en este experimento son cámaras 3D que emplean la tecnología de luz estructurada. Una básica y de uso doméstico, Kinect 360, que se muestra en la Figura 6(a), y otra cámara industrial, Zivid ONE+ M, que se puede ver en la Figura 6(b). De esta manera se va a hacer

(a) cámara básica Kinect 360 (b) cámara industrial Zivid ONE+ M

Figura 6: Cámaras 3D de luz estructurada usadas en la experimentación

un estudio del error que se produce en cada una de las cámaras cuando se aplica el método propuesto, comprobando si una cámara 3D industrial mejora los resultados de una cámara 3D básica, de uso más doméstico.

En cuanto al software de desarrollo, todos los cálculos y algoritmos del experimento se han implementado en lenguaje C#, bajo el *framework* Visual Studio.

Para este estudio no se ha usado la típica pieza de calibración que se suele emplear en otros experimentos basada en una pieza de ajedrez, como los autores por ejemplo usan en (Wang et al., 2020), sino que se ha empleado una estructura formada por 3 cubos unidos que es capaz de habilitar hasta quince puntos de calibración, los cuales son tomados en un orden determinado debido a que el algoritmo creado los ajusta acorde al orden de la toma de los mismos, como se mostraba en la Figura 2.

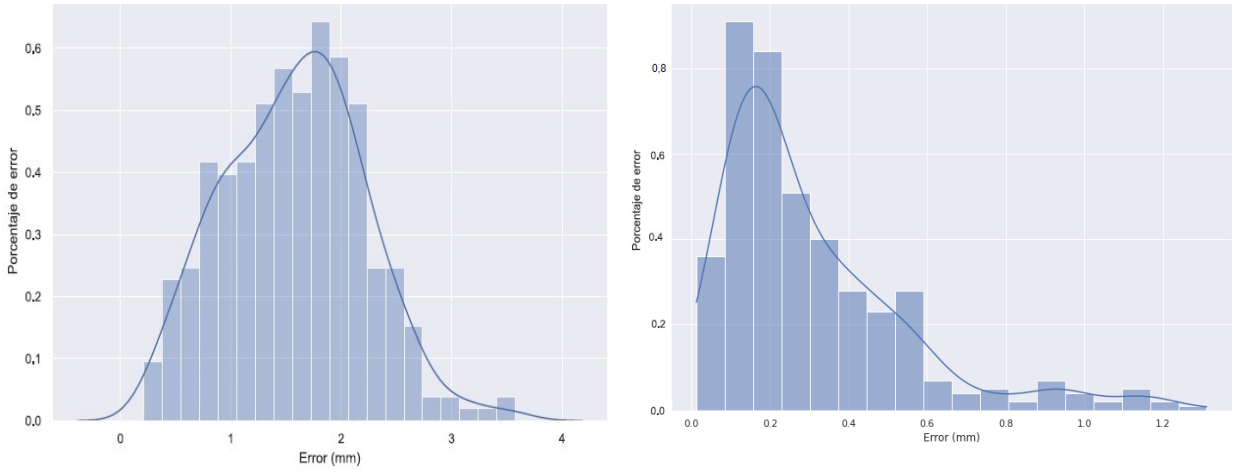
4.2. *Resultados y discusión*

El objetivo de la experimentación, además de obtener una matriz de rotación y traslación válida entre los puntos 3D capturados por la cámara y el brazo robótico, se encamina a realizar un estudio del error de repetibilidad del brazo robótico y del error producido en el proceso de calibración propuesto. En el caso de este último se comprobará cómo se comporta dicho proceso con una cámara 3D doméstica y con otra industrial.

En el caso del cálculo del error de repetibilidad del robot se tomaron los quince puntos de la pieza de calibración con la herramienta del robot ya calibrada, grabando cada uno de los puntos en un programa. Seguidamente se repitió el proceso un total de diez veces, es decir, se ejecutó el mismo programa nueve veces más, teniendo en total diez repeticiones del mismo, y por consiguiente, diez tomas de valores de cada uno de los quince puntos de calibración. A continuación, para comprobar si había algún tipo de punto que estuviera más separado que otro se tomó una aproximación de comparación (”todos con todos”. Es decir, se calcularon las distancias de los quince puntos, desde una repetición a todas las demás. Así, desde la repetición una se tomó la distancia a las repeticiones 2, 3, 4...10. Desde la repetición 2 a las repeticiones 3, 4, 5,...10. Así, hasta combinar la distancia entre puntos de cada una de las 10 repeticiones. Para el cálculo de esas distancias se tuvo en cuenta que siendo P_1 y P_2 los puntos a calcular sus distancias, tal que $P_1 = (x_1, y_1, z_1)$ y $P_2 = (x_2, y_2, z_2)$. La fórmula que se empleó para el cálculo de distancias fue la siguiente:

$$d(P_1, P_2 2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z1)^2}$$

En la Figura 9 se puede comprobar el error del robot producido en el experimento de repetibilidad. En el eje Y se observan los quince puntos de calibración, mientras que en el eje X están representadas cada una de las distancias de cada punto, desde cada repetición a las demás, según el enfoque que se ha comentado. Así, en total tenemos cuarenta y cinco mediciones de distancias, resultantes de cada uno de los quince puntos combinadas cada repetición con las demás. Se puede comprobar como el error obtenido no presenta ningún valor *outliner*, es decir, todos los valores de distancias entre puntos se encuentran



(a) Kinect 360 (b) Zivid ONE+ M

Figura 7: Porcentaje de acumulación de errores en milímetros entre las nubes de puntos capturadas con la cámara

agrupados en un rango de valores pequeño, por lo que podemos considerar la media con una métrica representativa de la muestra. Esta se sitúa en torno a las seis milésimas. Un resultado que en principio no tendría una incidencia destacable en el proceso de calibración presentado en este estudio.

Para medir el cálculo del error producido en el proceso de calibración se han realizado dos experimentos, de manera que se pueda obtener de ellos el error que comete la cámara y el error que se comete en el propio proceso planteado, al tomar los datos reales de la pieza de calibración, tras haber aplicado toda la metodología.

Para el cálculo del error acumulado en la cámara a la hora de capturar la escena en 3D se ha realizado una prueba de repetibilidad, es decir, se han tomado una serie de nubes de puntos de la pieza de calibración con cada una de las cámaras que intervienen en el estudio. Las capturas se han realizado con el algoritmo de fusión en el caso de la cámara básica, y con una captura simple en el caso de la cámara industrial. En total se tomaron un total de siete capturas seguidas, sin mover ni el robot, ni la pieza de calibración.

Para el método propuesto el error se define en este experimento como la distancia o diferencia entre el punto tomado como real (tomado con la herramienta del robot ya calibrada), con respecto al punto tomado/estimado con cada una de las cámaras.

Se compararon nubes dos a dos por cada punto de calibración, con el fin de medir el error que comete cada cámara al capturar la escena. Los datos se agruparon y se dispusieron en un histograma, para tener una visión más clara de los mismos y observar el error producido. En este caso, el histograma muestra la distribución de probabilidad de la acumulación de errores producidos en el estudio de la repetibilidad. En la Figura 7(a) se muestran los datos para la cámara Kinect 360, mientras que en la Figura 7(b), se muestra lo propio para la cámara Zivid. En el eje X se muestra el error en milímetros, mientras que el eje Y muestra la probabilidad, en tanto por uno, de la concentración del error en los puntos de calibración.

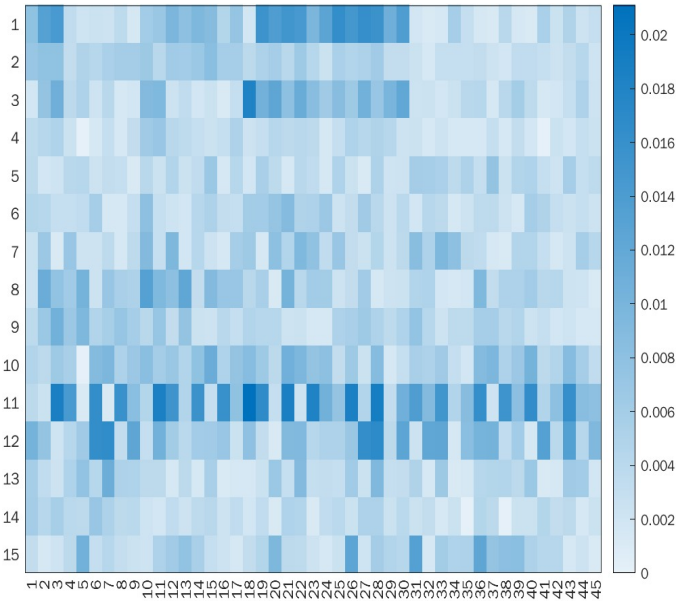


Figura 9: Distancia de cada una de las mediciones realizadas entre los quince puntos de calibración en las nueve repeticiones realizadas

Se observa cómo la cámara industrial genera un error muy inferior a la cámara básica en la captura de la escena 3D, ya que los datos con más probabilidad se encuentran sesgados claramente a la izquierda. Así, la probabilidad del error de captura de los puntos, aproximadamente un 80 %, se sitúa en torno a 0,1 - 0,2 milímetros. Igualmente, a simple vista se observa cómo los datos de probabilidad de la cámara básica se sitúan sin sesgo más orientados al centro del histograma, lo que nos hace indicar que la probabilidad del error producido es mayor. De esta manera, la agrupación principal de probabilidad de error se encuentra, aproximadamente un 60 %, en torno a 1,8 - 2 milímetros. Estos datos se deben a la fabricación de ambos dispositivos como pueden ser los parámetros intrínsicos de las cámaras, o la calidad y precisión de los componentes empleados en la manufactura.

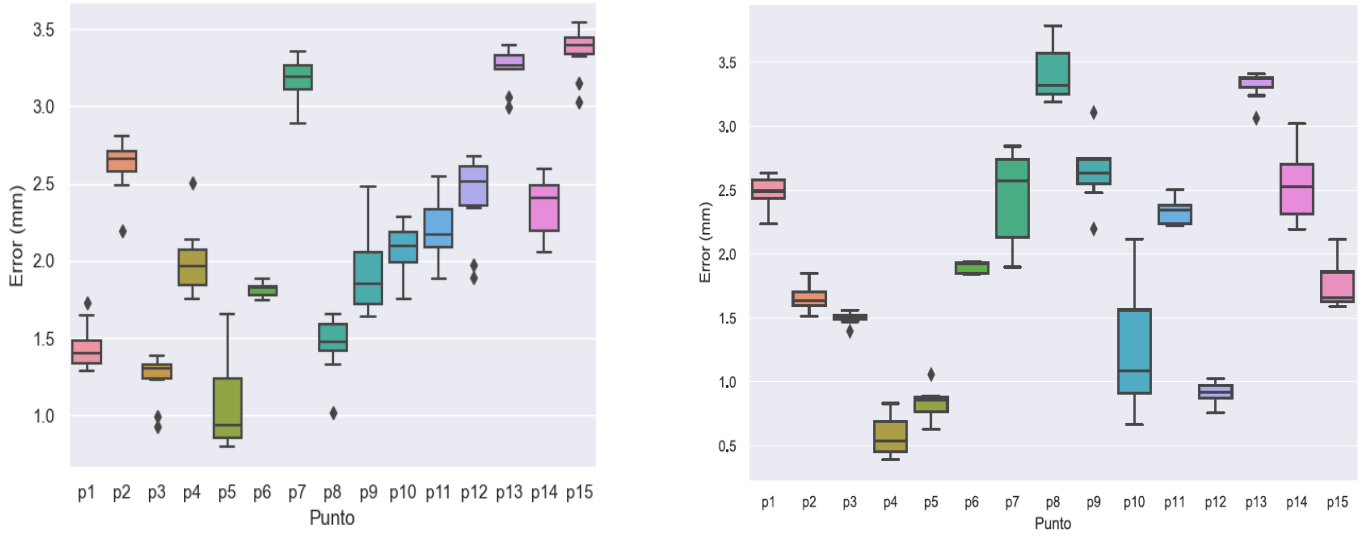
A continuación se ha medido la mejora en la precisión que la aplicación del algoritmo RANSAC aporta a la toma de datos manuales que requiere el proceso de calibración propuesto. Los puntos tomados manualmente son los mismos en la medición de cada cámara. Así, en la Figura 8(a) se muestran los errores de posición cometidos con la cámara Kinect 360, mientras que en la Figura 8(b) aparecen los mismos tomados con la cámara Zivid ONE+ M. Se puede observar cómo el error total medio se mejora más cuando se emplea la cámara industrial. Esto se puede explicar, igual que antes a que la fabricación del dispositivo influye en posteriormente en el error que pueda cometer, y en este caso en el ajuste que el algoritmo RANSAC aplique.

Desde la discusión de los resultados se quiere hacer mención al trabajo (Xu et al., 2019) donde se plasma un método de calibración similar al propuesto. La diferencia radica, en varios aspectos: inicialmente, el método propuesto es más sencillo de implantar y de seguir. En segundo lugar, en nuestro caso, se ha decidido emplear algoritmo RANSAC en lugar de la Transformada de Hough, porque entendemos que RANSAC ajustaría mejor la intersección de planos al hacerlo sobre miles de puntos obtenidos en el modelo 3D. En este caso se debería hacer un estudio sobre el método propuesto para verificar el resultado que se obtendría con la Transformada de Hough. Por último, en el trabajo mencionado se emplea un método para mejorar la precisión del sistema que proponen, *Bagging based Support Vector Machines* (BSVM) (Pham et al., 2018), mientras que en el trabajo propuesto en este artículo no se emplea ningún método o estrategia para mejorar la precisión de los puntos, que, en nuestro caso, proporciona RANSAC.

5. Conclusiones

En este estudio se ha presentado la fase inicial de un método de calibración que resuelve una calibración ojo a mano donde se obtiene el posicionamiento de un robot en función de la captura de la escena de una cámara 3D de luz estructurada. Para ello se contaba con que la cámara no estuviera en el efector final, por lo que se ha buscado un lugar idóneo para poder capturar el modelo 3D, estando anclada al robot.

El método se presenta como una solución fácil y rápida, ya que tiene como entrada un fichero de texto donde se almacenan las posiciones reales de los puntos de la pieza de calibración tomados manualmente con la herramienta que esté en el efector



(a) Kinect 360

(b) Zivid ONE+ M

Figura 8: Errores en milímetros de cada punto de la pieza de calibración por cada cámara tomados manualmente después de aplicar la metodología completa

final del robot (previamente calibrada), y una captura 3D de dicha pieza de calibración con la cámara de luz estructurada que se emplee. A partir de estos dos datos, el sistema propuesto nos devolverá una matriz de rotación y traslación de las coordenadas de la cámara a la base del robot. Para la obtención de estos resultados se emplea el algoritmo RANSAC, que detecta los puntos clave, que son los que se usan para posteriormente determinar las trayectorias. La intersección de planos se consigue utilizando miles de puntos estimados, lo cual permite realizar el cálculo en un solo paso (*single-step*), y no mediante un algoritmo iterativo como se ha visto en la bibliografía estudiada, lo que hace que el método se presente como una alternativa a los métodos actuales. No obstante, el sistema propuesto tiene margen de mejora, de maduración, y de incorporación de otras técnicas conocidas y aplicadas en otros métodos comentados durante la investigación.

En los experimentos realizados se obtiene como conclusión, de forma particular, que la cámara industrial ofrece un posicionamiento de fabricación mejor que la doméstica, cuestión comprobada en la prueba de repetibilidad. De la misma manera también la cámara industrial es capaz de ajustar mejor los errores del posicionamiento manual de los puntos de calibración después de aplicar el algoritmo RANSAC, por lo que se puede concluir que este algoritmo ajusta mejor los puntos, y por tanto disminuye más el error, cuando se parte de un error menor, como es el caso de la cámara Zivid ONE+ M. En el caso del robot empleado, FANUC LR Mate 20 iD, se observa un error de repetibilidad en torno a seis milésimas. En principio consideramos un error asumible en la mayoría de los trabajos. Este error habrá que tenerlo en cuenta, junto al error producido por la cámara en el proceso de calibración.

De forma general, se puede concluir que el hecho de emplear una cámara industrial de alta precisión va a disminuir notablemente el error final de posicionamiento, y por tanto merecería la pena usarla para cualquier tarea que precise un posicionamiento lo más exacto posible. También, en la misma medida es importante señalar que será determinante la elección de un robot industrial que tenga un error de repetibilidad tolerable por debajo del error que produce la medición de la cámara, por lo que la afectación al error de todo el proceso también es más baja.

Como trabajo futuro se propone realizar un estudio pormenorizado de otras fuentes de error que intervienen en el método presentado, con el fin de conocer mejor el impacto de cada una de esas fuentes de error para poder aplicar una corrección del mismo. Igualmente también sería interesante, realizar un análisis de la influencia del patrón de calibración en la precisión del método. aplicando diferentes distribuciones y con variadas cantidades de puntos. De forma general, también se propone estudiar la incorporación de cualquier acción, método o estrategia que aumentara la precisión de la calibración del método inicial propuesto.

Agradecimientos

Este trabajo ha sido realizado parcialmente gracias al apoyo del Plan Estatal de Investigación Científica y Técnica y de Innovación 2017-2020 (AUROVI) EQC2018-005190-P.

Fernando M. Quintana agradece al Ministerio de Ciencia, Innovación y Universidades de España su apoyo a través de la ayuda FPU (FPU18/04321).

Referencias

Ali, I., Suominen, O., Gotchev, A., Morales, E. R., jun 2019. Methods for simultaneous robot-world-hand–eye calibration: A comparative study. Sensors (Switzerland) 19 (12), 2837.
DOI: 10.3390/s19122837

Cui, H., Sun, R., Fang, Z., Lou, H., Tian, W., Liao, W., 2020. A novel flexible two-step method for eye-to-hand calibration for robot assembly system. Measurement and Control 53 (9-10), 2020–2029.
DOI: 10.1177/0020294020964842

de Klerk, E., 2002. Aspects of semidefinite programming interior point algorithms and selected applications. Applied Optimization.

Derpanis, K. G., 2010. Overview of the RANSAC Algorithm. Tech. rep., EECS.

Dias, J., de Almeida, A., Araujo, H., Batista, J., 1991. Improving camera calibration by using multiple frames in hand-eye robotic systems .
DOI: 10.1109/IROS.1991.174464

Eriksson, T., Hansen, H. N., Gegeckaite, A., 2008. On the use of industrial robots in microfactories. The International Journal of Advanced Manufacturing Technology 38 (5), 479–486.
DOI: 10.1007/s00170-007-1116-7

Featherstone, R., 2007. Robot dynamics algorithms .

Fischler, M., Bolles, R., 1981.
DOI: 10.1145/358669.358692

Hu, J.-S., Chang, Y.-J., 2013. Automatic Calibration of Hand-Eye-Workspace and Camera Using Hand-Mounted Line Laser .
DOI: 10.1109/TMECH.2012.2212717

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., Fitzgibbon, A., 2011. KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In: UIST '11 Proceedings of the 24th annual ACM symposium on User interface software and technology, uist '11 proceedings of the 24th annual acm symposium on user interface software and technology Edition. ACM, pp. 559–568.

Koide, K., Menegatti, E., 2019. General Hand-Eye Calibration Based on Reprojection Error Minimization .
DOI: 10.1109/LRA.2019.2893612

Lasi, H., Kemper, H.-G., Feld, D.-I. T., Hoffmann, D.-H. M., 2014. Industry 4.0. Business & Information Systems Engineering 4, 239–242.
DOI: 10.1007/s12599-014-0334-4

Li, W., Dong, M., Lu, N., Lou, X., Sun, P., 2018. Simultaneous Robot-World and Hand-Eye Calibration without a Calibration Object .
DOI: 10.3390/s18113949

Liu, X., Madhusudanan, H., Chen, W., Li, D., Ge, J., Ru, C., Sun, Y., 2021. Fast eye-in-hand 3-d scanner-robot calibration for low stitching errors. IEEE Transactions on Industrial Electronics 68 (9), 8422–8432.
DOI: 10.1109/TIE.2020.3009568

Low, K.-L., 2004. Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration. Tech. rep., University of North Carolina.

Lu, Y., jun 2017. Industry 4.0: A survey on technologies, applications and open research issues.
DOI: 10.1016/j.jii.2017.04.005

Lundberg, I., Bjorkman, M., Ogren, P., 2014. Intrinsic camera and hand-eye calibration for a robot vision system using a point marker .
DOI: 10.1109/HUMANOIDS.2014.7041338

Magnenat-Thalmann, N., 2020. Preface the Visual Computer (vol 36 issues 10–12) .
DOI: 10.1007/s00371-020-01965-8

Munchen, T. U., 2009. HandEyeCalibration.
URL: http://campar.in.tum.de/Chair/HandEyeCalibration

Newcombe, R. A., Fitzgibbon, A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S., 2011. KinectFusion: Real-time dense surface mapping and tracking .
DOI: 10.1109/ISMAR.2011.6092378

Pachtrachai, K., Vasconcelos, F., Chadebecq, F., Allan, M., Hailes, S., Pawar, V., Stoyanov, D., 2018. Adjoint Transformation Algorithm for Hand–Eye Calibration with Applications in Robotic Assisted Surgery .
DOI: 10.1007/s10439-018-2097-4

Pham, B. T., Tien Bui, D., Prakash, I., 2018. Bagging based Support Vector Machines for spatial prediction of landslides .
URL: https://go.exlibris.link/8LbXd1pD
DOI: 10.1007/s12665-018-7268-y

Rusinkiewicz, S., Levoy, M., 2001. Efficient variants of the ICP algorithm. Proceedings of International Conference on 3-D Digital Imaging and Modeling, 3DIM, 145–152.
DOI: 10.1109/IM.2001.924423

Shiu, Y. C., Ahmad, S., 1989. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB .
DOI: 10.1109/70.88014

Sorkine-Hornung, O., Rabinovich, M., 2017. Least-Squares Rigid Motion Using SVD. Tech. rep., Department of Computer Science, ETH Zurich.
URL: http://www.igl.ethz.ch/projects/ARAP/

Taryudi, Wang, M.-S., 2018. Eye to hand calibration using ANFIS for stereo vision-based object manipulation system .
DOI: 10.1007/s00542-017-3315-y

Toan, N. V., Khoi, P. B., 2018. A svd-least-square algorithm for manipulator kinematic calibration based on the product of exponentials formula †. Journal of Mechanical Science and Technology 32 (11), 5401–5409.
DOI: 10.1007/s12206-018-1038-3

Tsai, R. Y., Lenz, R. K., 1989. A New Technique for Fully Autonomous and Efficient 3D Robotics Hand/Eye Calibration. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION 5 (3).

Uhlig, F., 1992. Review of topics in matrix analysis .
DOI: 10.1016/0024-3795(92)90075-L

Wang, Z., Fan, J., Jing, F., Deng, S., Zheng, M., Tan, M., 2020. An Efficient Calibration Method of Line Structured Light Vision Sensor in Robotic Eye-in-Hand System .
DOI: 10.1109/JSEN.2020.2975538

Werner, D., Al-Hamadi, A., Werner, P., 2014. Truncated Signed Distance Function: Experiments on Voxel Size .
DOI: 10.1007/978-3-319-11755-3-40

Xu, F., Fan, S., Yang, Q., Zhang, C., Wang, Y., 2019. Welding robotic hand-eye calibration method based on structured light plane .
URL: https://go.exlibris.link/WK5ldmt3
DOI: 10.23919/ChiCC.2019.8865169

Yang, M. Y., Förstner, W., 2010. Plane Detection in Point Cloud Data. Tech. rep., Department of Photogrammetry Institute of Geodesy and Geoinformation University of Bonn.
URL: http://www.ipb.uni-bonn.de/technicalreports/

Zhang, Z., Zhang, L., Yang, G.-Z., 2017. A computationally efficient method for hand–eye calibration .
DOI: 10.1007/s11548-017-1646-x

Zhao, Z., Liu, Y., 2009. A hand–eye calibration algorithm based on screw motions .
DOI: 10.1017/S0263574708004608

Zou, Y., Chen, X., 2018. Hand–eye calibration of arc welding robot and laser vision sensor through semidefinite programming .
DOI: 10.1108/IR-02-2018-0034