

Calibración Extrínseca de un Conjunto de Cámaras RGB-D sobre un Robot Móvil

David Zúñiga-Noël, Rubén Gómez Ojeda, Francisco-Ángel Moreno, Javier González Jiménez
 Departamento de Ingeniería de Sistemas y Automática, grupo MAPIR, Universidad de Málaga
 Instituto de Investigación Biomédica de Málaga (IBIMA)
 {dzuniga,rubengooj,famoreno,javiergonzalez}@uma.es

Resumen

La aparición de las cámaras RGB-D como sensores robóticos de bajo coste ha supuesto la inclusión habitual de varios de estos dispositivos en una creciente cantidad de vehículos y robots. En estos casos, la calibración precisa de las transformaciones espaciales existentes entre las cámaras del mismo robot es de capital importancia a la hora de obtener medidas fiables del entorno. Este artículo evalúa el método de calibración con fórmula cerrada descrito en [7] y lo extiende con una propuesta alternativa basada en un método iterativo y una extensión robusta de este último en dos escenarios: i) un entorno simulado con cambios en el nivel de ruido de las observaciones, en el número de observaciones obtenidas, en la proporción de outliers y en las posiciones relativas de las cámaras, y ii) una configuración particular de 3 cámaras RGB-D sobre un robot real. Los resultados de la evaluación muestran una mayor precisión para nuestra propuesta iterativa robusta en todos los escenarios analizados. El código fuente de la implementación de estos métodos en C++ se proporciona para su uso público.

Palabras clave: Calibración extrínseca, cámaras RGB-D, robots móviles.

1. Introducción

La aparición de las cámaras RGB-D ha supuesto una revolución en el mundo de los sensores robóticos debido a su bajo coste y la gran cantidad de información que pueden proporcionar sobre el entorno, combinando una cámara estándar RGB con una cámara especial (principalmente de proyección de patrones de infrarrojos o de tiempo de vuelo) que captura la información de profundidad de la escena.

Siendo una de las principales limitaciones de este tipo de dispositivos su relativamente bajo campo de visión ($\sim 60^\circ$ HFoV), es habitual encontrar sistemas que integran varias cámaras RGB-D funcionando simultáneamente sobre un mismo robot o vehículo [4, 13], aumentando de esta forma su rango de observación. Este tipo de configuraciones

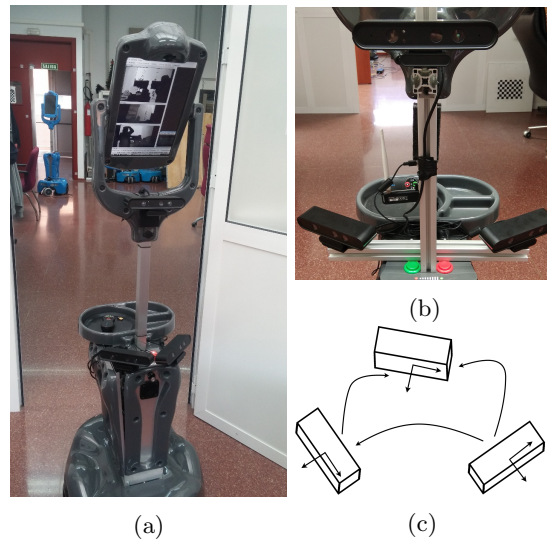


Figura 1: Figura con el robot *Giraff* y el conjunto de sensores RGB-D con los que se han llevado a cabo los experimentos (Figura 1a). En la Figura 1b se puede observar en detalle las tres cámaras montadas en el sensor, mientras que en la Figura 1c se puede observar el esquema del conjunto de cámaras con sus transformaciones relativas.

multi-cámara implican un proceso de calibración extrínseca de las mismas que determina su posición y orientación dentro de un sistema de referencia común elegido arbitrariamente al que se transforman las medidas tomadas por todas las cámaras, generando una observación precisa y coherente del entorno.

Es importante remarcar que esta calibración extrínseca de las configuraciones multi-cámara es independiente del proceso de calibración intrínseco (e.g. estimación de la distancia focal, parámetros de distorsión, etc.) de cada una de ellas, así como de la estimación de la transformación espacial existente entre la cámara RGB y de distancia que componen cada dispositivo individual [18, 6, 8]. Además, las cámaras RGB-D sufren de la aparición de errores en las medidas de rango en función de la distancia de los objetos medidos al centro óptico de la cámara de profundidad, requiriendo de un tratamiento especial para compensar dichos errores y obtener medidas fiables [14, 5].

Estos procedimientos de calibración son de capital importancia para obtener medidas precisas del entorno, reduciendo de esta manera los errores sistemáticos en las observaciones obtenidas.

En este artículo, sin embargo, asumimos que las cámaras ya se encuentran calibradas individualmente y nos centramos principalmente en la estimación robusta de las transformaciones geométricas existentes entre un conjunto de sensores RGB-D que forman un sólido rígido. Para ello extendemos el método propuesto en [7], que implementa una formulación cerrada para el caso específico de dos cámaras. Dicho método está basado en la observación de superficies planas desde diferentes perspectivas y tiene la ventaja de que puede ser aplicado de forma sencilla sin necesidad de un patrón de calibración pre-construido, ya que, habitualmente, los entornos creados por seres humanos presentan dichas superficies. De esta forma, el proceso de re-calibración puede ejecutarse en cualquier momento y prácticamente sin intervención humana. Sin embargo, la formulación cerrada propuesta se muestra muy sensible ante la presencia de medidas afectadas por mucho ruido y/o observaciones erróneas (*outliers*), por lo que se hace necesario el desarrollo de un método que pueda hacer frente de mejor manera a estas situaciones habituales en los experimentos con datos reales.

Así, las contribuciones de este trabajo se concretan en:

- La elaboración de un estudio de los resultados obtenidos por el método propuesto en dicho artículo, tanto en simulación como en un robot real.
- La propuesta de un método alternativo basado en una formulación iterativa y una versión robusta del mismo que obtiene mejores resultados incluso en presencia de *outliers* y/o medidas muy ruidosas.
- La publicación del código fuente en C++ que implementa los tres métodos evaluados.

Para la validación de los resultados, se han desarrollado dos tipos de pruebas (ver sección 5): i) en un entorno simulado en el que se ha analizado el comportamiento de los distintos sistemas propuestos e implementados ante distintos niveles de ruido, número de observaciones, proporción de *outliers* en las observaciones y posiciones relativas de los sensores y ii) con una configuración de tres cámaras sobre un robot real donde se han estimado las transformaciones relativas entre cada par de cámaras y se han comparado con las obtenidas por la composición de las otras dos transformaciones. En todos los casos se ha demostrado que el

método iterativo robusto presenta una mayor precisión a la hora de determinar la posición relativa entre cámaras.

Por último, el código fuente de las librerías de C++ desarrolladas en este artículo, junto con vídeos ilustrativos y demás material adicional, podrá ser encontrado en el siguiente enlace: <http://mapir.uma.es/>

2. Trabajos Relacionados

La necesidad de métodos de calibración extrínseca de sensores ha dado lugar a diversas propuestas, tanto genéricas [15] como para sensores específicos [8]. La mayoría de estos métodos de calibración pueden clasificarse en dos grandes grupos: *basados en el movimiento* o *basados en la observación simultánea de características*.

Los métodos *basados en movimiento* [15, 3, 19] consisten básicamente en estimar la relación espacial entre los sensores a partir de la trayectoria calculada para cada sensor independientemente. Si bien se trata de un enfoque genérico, puesto que únicamente requiere un algoritmo de odometría, puede ser una solución poco práctica. Una de las principales complicaciones que se presentan a la hora de aplicar estos métodos de manera automática es que el resultado de la calibración depende directamente de la precisión del algoritmo de odometría en que se base. Además, la precisión puede variar de un entorno a otro y según la cantidad de elementos dinámicos o estáticos presentes.

Por el contrario, los *métodos basados en la observación simultánea de características* [10, 17, 8] consisten en detectar correspondencias entre elementos del entorno percibidos simultáneamente por los sensores, a partir de las cuales se estima la relación espacial entre dichos sensores. En general, estos métodos, a diferencia de los basados en movimiento, necesitan de una región de observación común a los sensores para poder establecer las correspondencias. Además, las características del entorno en base a las que se formulan estas soluciones deben aparecer con suficiente frecuencia para poder realizar el procedimiento de calibración de manera autónoma en un robot móvil.

El método [7] en el que se basa este trabajo entraría dentro de esta última categoría, empleando regiones planas como características sobre las que establecer las correspondencias. Este planteamiento, en caso de que las regiones sean suficientemente grandes, relaja la restricción de *observación simultánea* a la observación de un mismo plano, pero no necesariamente la misma región del plano. Esta característica permite minimizar la región de solape entre los sensores, maximizando de esta for-

ma el campo de visión. Además, la abundancia de regiones planas en entornos creados por el ser humano permite asegurar la presencia de las características en el entorno de trabajo necesarias para llevar a cabo el proceso de calibración.

Sin embargo, como ya hemos comentado, el planteamiento original [7] se ve afectado considerablemente por el ruido de las medidas y la presencia de *outliers* en las observaciones. Para mitigar estos efectos, en este trabajo se aplican técnicas de optimización robusta que permiten obtener mayor precisión en estas condiciones.

3. El Robot *Giraff*

Giraff es un robot móvil diseñado con fines asistenciales que está basado en una plataforma robótica con un PC a bordo y una pantalla táctil ajustable en altura y orientación que se utiliza como principal interfaz de interacción con el usuario (ver Figura 1a).

3.1. Hardware

La plataforma robótica consta de dos ruedas motoras y dos ruedas libres situadas una en la parte delantera y otra en la trasera con el fin de mejorar la estabilidad del movimiento y permitir que el robot gire sobre sí mismo. El conjunto de sensores del robot se compone de una cámara web de 5 Mpx situada en la parte superior de la pantalla táctil con la que se puede transmitir video a un usuario remoto o detectar y reconocer tanto personas como objetos. En la parte delantera del robot se han colocado dos cámaras RGB-D con el objetivo de ampliar el campo de visión del robot para el mapeado, la localización y la navegación. La posición de las cámaras se ha escogido de manera que las vistas de ambos sensores no se solapen pero que tengan una visión del suelo en todo momento. Otra cámara RGB-D situada a una altura mayor se utilizará para el reconocimiento de gestos, como método alternativo de interacción entre el usuario y el robot.

3.2. Software

El PC de a bordo de *Giraff* está basado en Linux Ubuntu 16.04 e integra un entorno ROS donde se implementan las funcionalidades y los comportamientos del robot: navegación autónoma, detección y reconocimiento de caras y gestos, análisis y síntesis de voz, etc. Dentro de esta arquitectura se integra el módulo de auto-calibración que implementa el método descrito en este artículo.

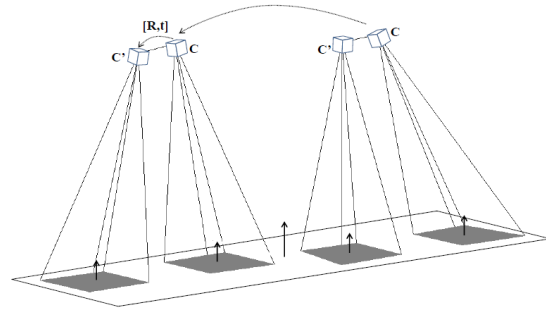


Figura 2: Figura adaptada de [7] con el esquema de calibración empleado en este trabajo, en el que dos sensores RGB-D unidos rígidamente por una transformación $T = [R|t]$ observan en distintas configuraciones el mismo plano.

4. Calibración Extrínseca de Cámaras RGB-D

En esta sección se describen los algoritmos de calibración desarrollados en este artículo, tras introducir brevemente el problema que vamos a resolver.

4.1. Planteamiento del Problema

Esencialmente, el objetivo es encontrar la transformación geométrica relativa $T = [R|t] \in SE(3)$ entre dos sensores RGB-D, suponiendo que están unidos rígidamente (ver Figura 2). Para ello, se obtendrá una serie de N distintas observaciones de uno o más planos, dadas por su vector normal \mathbf{n}_i en el sistema de referencia de cada cámara y la distancia desde el centro óptico de éstas al plano d_i . Una vez obtenidas las observaciones, suponiendo que cumplen la condición de observabilidad, dada por:

$$\text{rango}\left(\sum_{i=1}^N \mathbf{n}_i \mathbf{n}_i^T\right) = 3 \quad (1)$$

se puede encontrar la solución óptima al problema (esto es, estimar la pose relativa entre los diferentes sensores) de diferentes maneras. Para que se cumpla la condición de observabilidad, al menos 3 observaciones independientes son necesarias [7]

4.2. Solución en Forma Cerrada

En este apartado, se describirá someramente la solución en forma cerrada al problema planteado presentada previamente en [7] que desacopla la estimación de la traslación con respecto de la rotación, comenzando por la estimación de la rotación \mathbf{R} . Para ello, el estimador de máxima verosimilitud viene dado por la maximización de la verosimilitud logarítmica, asumiendo que los errores

siguen una distribución normal, para las N observaciones:

$$\operatorname{argmax}_{\mathbf{R}} \left\{ \log \prod_{i=1}^N p(\mathbf{n}_i, \mathbf{n}'_i | \mathbf{R}) \right\} \quad (2)$$

donde las primas en los superíndices denotan observaciones del segundo sensor. Entonces, gracias a la hipótesis de que los errores son Gaussianos, podemos llegar a la siguiente expresión:

$$\operatorname{argmin}_{\mathbf{R}} \sum_{i=1}^N \omega_i \|\mathbf{n}_i - \mathbf{R}\mathbf{n}'_i\|^2 \quad (3)$$

observando que la derivación del estimador de máxima verosimilitud coincide con el problema de mínimos cuadrados justamente introducido. La ecuación anterior es también conocida como problema ortogonal de Procrustes y su solución en forma cerrada es conocida [1, 16].

En el caso de la traslación, es posible llegar fácilmente a la siguiente expresión:

$$\operatorname{argmin}_{\mathbf{t}} \left\{ \omega_i (d_i - d'_i + \mathbf{t} \cdot \mathbf{R} \cdot \mathbf{n}'_i)^2 \right\} \quad (4)$$

que, al igual que en el caso anterior, corresponde el estimador de máxima verosimilitud:

$$\operatorname{argmax}_{\mathbf{t}} \left\{ \log \prod_{i=1}^N p(\mathbf{n}_i, \mathbf{n}'_i, d_i, d'_i, \mathbf{R} | \mathbf{t}) \right\} \quad (5)$$

con el problema de mínimos cuadrados presentado, cuya solución es sencilla de recuperar (consultar [7] para más detalles).

4.3. Solución con Mínimos Cuadrados

El problema de la solución anteriormente presentada, llega en presencia de observaciones muy ruidosas y/o *outliers*, ya que es muy sensible a estos fenómenos. Por ello, en este artículo proponemos su resolución de forma robusta mediante los problemas de mínimos cuadrados planteados en (3) y (4). Escribiendo dichas ecuaciones en una misma expresión de forma matricial, se puede llegar a:

$$\operatorname{argmin}_{\boldsymbol{\xi}} \left\{ \left[\begin{array}{c} \mathbf{n}_i \\ d_i - d'_i \end{array} \right] - \left[\begin{array}{c} \mathbf{R} \\ -\mathbf{t}^T \mathbf{R} \end{array} \right] \mathbf{n}'_i \right\}^2 \quad (6)$$

donde $\boldsymbol{\xi} \in \mathfrak{se}(3)$ es el vector que define la transformación relativa en la variedad 6-dimensional $\mathfrak{se}(3)$, conocida como álgebra de Lie, del espacio de las transformaciones Euclídeas de forma que se cumple $\mathbf{T}(\boldsymbol{\xi}) \in SE(3)$.

Si se observa la segunda parte de la expresión (6), es posible relacionarla con la traspuesta de la inversa de la matriz de transformación 3×4 existente

entre los dos sensores, esto es:

$$\boldsymbol{\xi}^* = \operatorname{argmin}_{\boldsymbol{\xi}} \left\{ \left[\begin{array}{c} \mathbf{n}_i \\ d_i - d'_i \end{array} \right] - \mathbf{T}(\boldsymbol{\xi})^{-T} \mathbf{n}'_i \right\}^2 \quad (7)$$

donde $\boldsymbol{\xi}^*$ se refiere al valor óptimo estimado. En lugar de resolver la expresión anterior de forma cerrada, se va a resolver el problema mediante el algoritmo iterativo Levenberg-Marquardt [11] de común uso en robótica [9]. Dicho algoritmo requiere de la estimación de la matriz Jacobiana del error minimizado, o dicho de otro modo:

$$\mathbf{J}(\boldsymbol{\xi}) = \left. \frac{\partial \mathbf{e}_i(\boldsymbol{\xi} \oplus \Delta \boldsymbol{\xi})}{\partial \Delta \boldsymbol{\xi}} \right|_{\Delta \boldsymbol{\xi} = 0} \quad (8)$$

donde el operador \oplus es la composición de poses y el error \mathbf{e}_i se refiere a:

$$\mathbf{e}_i(\boldsymbol{\xi}) = \left[\begin{array}{c} \mathbf{n}_i \\ d_i - d'_i \end{array} \right] - \mathbf{T}(\boldsymbol{\xi})^{-T} \mathbf{n}'_i \quad (9)$$

tras lo cual se puede obtener el valor óptimo de la transformación relativa entre los dos sensores $\boldsymbol{\xi}^*$. Para mayor detalle en la matemática relacionada con la geometría Euclídea recomendamos al lector consultar [2].

Por último, similarmente al método presentado en [12], se puede mejorar el comportamiento de la solución (7) ante observaciones ruidosas y/o *outliers*. Para ello, en lugar de utilizar la norma Euclídea, muy sensible a los grandes errores introducidos por las observaciones ruidosas, se ha utilizado la siguiente función robusta:

$$\boldsymbol{\xi}^* = \operatorname{argmin}_{\boldsymbol{\xi}} \rho(\|\mathbf{e}_i(\boldsymbol{\xi})\|^2) \quad (10)$$

donde $\rho(s) = \log(1 + s)$. En consecuencia, la matriz Jacobiana en (8) puede ser expresada como:

$$\mathbf{J}(\boldsymbol{\xi}) = \left. \frac{\partial \rho(\|\mathbf{e}_i(\boldsymbol{\xi} \oplus \Delta \boldsymbol{\xi})\|^2)}{\partial \mathbf{e}_i(\boldsymbol{\xi} \oplus \Delta \boldsymbol{\xi})} \frac{\partial \mathbf{e}_i(\boldsymbol{\xi} \oplus \Delta \boldsymbol{\xi})}{\partial \Delta \boldsymbol{\xi}} \right|_{\Delta \boldsymbol{\xi} = 0} \quad (11)$$

que se puede relacionar fácilmente con el Jacobiano estimado anteriormente en (8).

5. Experimentos

Con el objetivo de analizar y comparar la solución original con las modificaciones propuestas en este trabajo, se ha llevado a cabo una serie de experimentos empleando tanto datos simulados (Sección 5.1) como datos reales (Sección 5.2) obtenidos mediante cámaras RGB-D montadas sobre el robot *Giraff*.

5.1. Entorno Simulado

Para la evaluación con datos simulados, se ha generado un conjunto de $N = 100$ transformaciones conocidas $T \in SE(3)$ entre los dos sensores, variando la traslación en el rango $[0.5, 1.5]$ m y la rotación en el rango $[-45, 45]$ °, ambos en el álgebra de Lie $\mathfrak{se}(3)$. Las observaciones en el espacio se han obtenido distribuyendo uniformemente los valores de traslación en ± 3 m y rotación en $\pm 180^\circ$, para uno de los sensores, de forma que se cumpla la condición de observabilidad (1).

Para evaluar el funcionamiento del algoritmo presentado ante el máximo número de fenómenos posibles en casos reales, se han simulado variaciones en los siguientes parámetros:

- Número de observaciones de planos N_{obs} .
- Nivel de ruido del sensor K_σ .
- Porcentaje de *outliers* R_{out} ,

en los rangos presentados en las Figuras 3, 4 y 5. Cada experimento, dada una configuración de sensores, se ha repetido un número de 100 veces dando lugar a un total de 10000 simulaciones para cada configuración de parámetros $\{N_{obs}, K_\sigma, R_{out}\}$, teniendo en cuenta que los parámetros por defecto son $N_{obs} = 100$, $K_\sigma = 1$, y $R_{out} = 10\%$.

En el caso del ruido de sensor, se ha considerado a éste como Gaussiano, cuyo nivel varía en función de K_σ de la siguiente manera:

$$\mathbf{n} \sim \mathcal{N}(\boldsymbol{\mu}_n, K_\sigma \Sigma_n), \quad d \sim \mathcal{N}(\boldsymbol{\mu}_d, K_\sigma \sigma_d^2) \quad (12)$$

donde $\boldsymbol{\mu}_n$ y $\boldsymbol{\mu}_d$ son los parámetros medios de observación del plano y Σ_n y σ_d son, respectivamente, estimaciones de la matriz de covarianza y desviación típica asociadas a los parámetros, empleando para ello valores representativos obtenidos empíricamente.

Los *outliers*, por otra parte, se han simulado introduciendo observaciones sin relación *a priori* con los parámetros de calibración extrínseca de los sensores. Para ello, se han sustituido los parámetros de una de las observaciones que forman cada correspondencia con parámetros de un plano extraídos de una distribución uniforme en todo el espacio de posibles observaciones considerado durante la simulación.

Para evaluar los algoritmos presentados en este trabajo, se ha empleado la distancia en la variedad 6-dimensional $\mathfrak{se}(3)$ entre las pose estimada y real, $\boldsymbol{\xi}^*$, $\boldsymbol{\xi}_r \in \mathfrak{se}(3)$ respectivamente, esto es:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{\omega} \end{bmatrix} = \boldsymbol{\xi}^* \ominus \boldsymbol{\xi}_r = \log\left(\exp(\boldsymbol{\xi}^*)^{-1} \cdot \exp(\boldsymbol{\xi}_r)\right) \quad (13)$$

donde $\boldsymbol{\tau}$ y $\boldsymbol{\omega}$ son respectivamente las partes de traslación y rotación del vector $\boldsymbol{\varepsilon} \in \mathfrak{se}(3)$, y las funciones $\exp : \mathfrak{se}(3) \mapsto SE(3)$ y $\log : SE(3) \mapsto \mathfrak{se}(3)$ se encargan de relacionar el álgebra de Lie $\mathfrak{se}(3)$ y el espacio Euclídeo $SE(3)$.

En la Figura 3 se muestran los errores relativos tanto en rotación como en traslación (relativos a la transformación real; en tanto por uno) para un número variable de observaciones N_{obs} . En la gráfica se observa una ligera mejora en la precisión del algoritmo Levenberg-Marquardt robusto (LM-R) con respecto a la solución en forma cerrada (FC). La tendencia de los errores se vuelve asintótica a medida que aumenta el número de observaciones, esto es, a partir de un número de observaciones deja de ser beneficioso introducir nuevas medidas al estar incurriendo en un coste computacional más elevado para una mejora marginal en la precisión.

En cambio, en la Figura 4 se representan los errores relativos para un diferentes niveles de ruido K_σ , donde podemos observar que la solución con peor comportamiento es FC, seguida de LM y, de nuevo, la solución con mejor precisión vuelve a ser LM-R. Además, se puede observar como la solución FC empeora más que las que emplean Levenberg-Marquardt (LM y LM-R) a medida que aumenta el nivel de ruido en el sensor.

Por último, la Figura 5 presenta los errores para un porcentaje variable de *outliers*. En esta gráfica se puede observar de forma intuitiva cómo los algoritmos que no emplean ningún método para mejorar su robustez empeoran su comportamiento dramáticamente en función de la presencia de *outliers*, mientras que el método Levenberg-Marquardt robusto (LM-R) es capaz de realizar la calibración con un error entorno al 10%, tanto en traslación como en rotación, para un nivel de *outliers* del 50%.

5.2. Sobre el Robot *Giraff*

Finalmente, se ha evaluado el funcionamiento de los métodos analizados mediante un experimento en el cual se calibra extrínsecamente un conjunto de tres cámaras RGB-D montado sobre la plataforma robótica *Giraff* (ver Figura 1). Dicho experimento se llevó a cabo en un entorno de oficina, en el que se ha movido al robot de forma manual tomando como planos de referencia el suelo y las paredes, eludiendo de esta forma la necesidad de utilizar un patrón de calibración construido para tal efecto.

Dado que en este experimento no se dispone de la transformación real que actúe como *ground truth*, se han calculado las transformaciones relativas en-

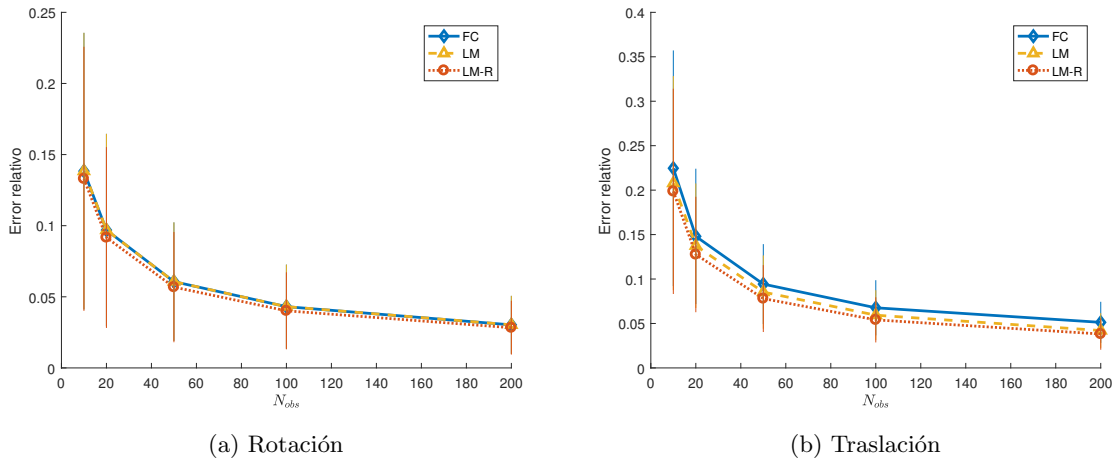


Figura 3: Resultados obtenidos en simulación variando el número de observaciones N_{obs} , para el error en la rotación (4a) y en la traslación (4b), expresado en términos de error relativo.

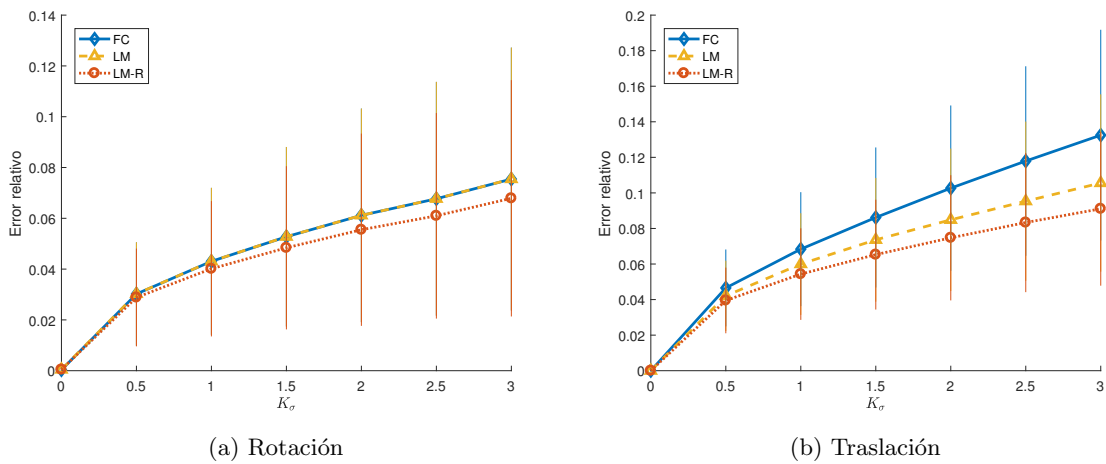


Figura 4: Resultados de simulación obtenidos variando el nivel de ruido K_σ , para el error en la rotación (4a) y en la traslación (4b), expresado en términos de error relativo.

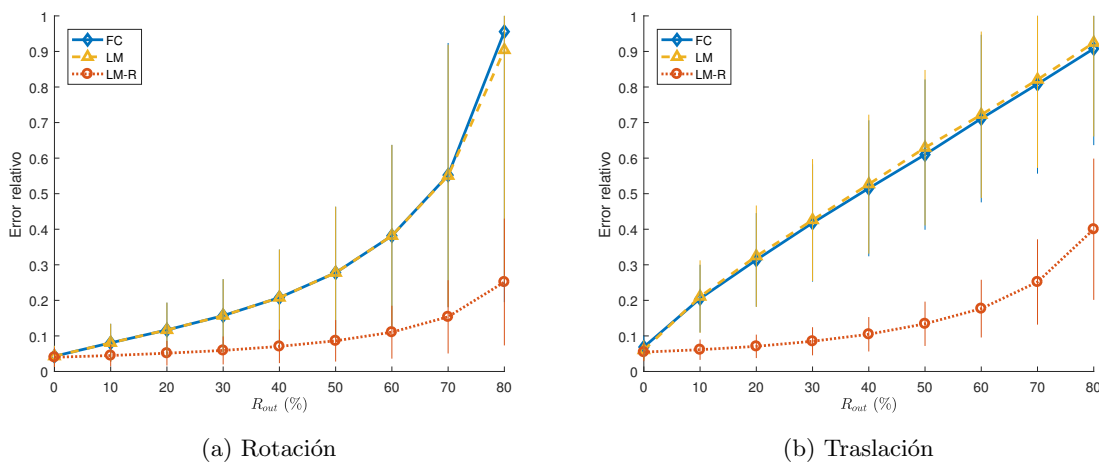


Figura 5: Resultados obtenidos variando la proporción de *outliers* R_{out} , para el error en la rotación (4a) y en la traslación (4b), expresado en términos de error relativo.

tre pares de cámaras ($\xi_0^1, \xi_0^2, \xi_2^1$) y se han comparado las transformaciones estimadas con las obtenidas mediante la composición de las otras dos transformaciones relativas, calculando el error del cierre de bucle mediante:

$$\varepsilon = (\xi_0^2 \oplus \xi_2^1) \ominus \xi_0^1 \quad (14)$$

En la Tabla 1 se muestran los errores obtenidos al evaluar los parámetros de calibración mediante el procedimiento descrito anteriormente. Si bien los tres métodos presentan errores similares, el método LM-R se ha empleado sin ninguna estimación inicial de los parámetros, mientras que en los otros dos métodos precisan una estimación inicial para descartar falsas correspondencias entre planos observados.

Tabla 1: Errores en el cierre de bucle tras estimar las transformaciones por parejas y medir su diferencia.

	τ_{err}	ω_{err}
FC	0.06 m	7.92°
LM	0.04 m	7.3°
LM-R	0.04 m	7.02°

6. Conclusiones

En este artículo se ha abordado el problema de la calibración extrínseca de un conjunto de cámaras RGB-D que forman un sólido rígido mediante el método de formulación cerrada propuesto en [7], un método basado en una solución iterativa y la extensión de este último con una formulación robusta en presencia de *outliers* y medidas altamente ruidosas.

Para ello se ha realizado un estudio comparativo de los errores cometidos por los tres métodos en un entorno simulado con condiciones cambiantes de niveles de ruido, número de observaciones, proporción de *outliers* en las observaciones y posiciones relativas de los sensores. Los experimentos muestran una mayor precisión de nuestra variante robusta del método iterativo en todos los escenarios analizados.

En un segundo experimento, los tres métodos han sido evaluados para una configuración de 3 cámaras RGB-D montadas en un robot real moviéndose en un entorno de oficina. En este experimento se han estimado las transformaciones relativas entre cada par de cámaras y se ha calculado el error existente entre una de las transformaciones estimadas y la obtenida al componer las otras dos. De igual forma, el método iterativo robusto ha demostrado obtener mejores resultados a la hora de estimar las transformaciones de las cámaras.

El código fuente de las librerías de C++ desarrolladas en este artículo ha sido publicado y podrá encontrarse en <http://mapir.uma.es>

Referencias

- [1] Arun, K. S., Huang, T. S., and Blostein, S. D. (1987). Least-squares fitting of two 3-D point sets. *IEEE Transactions on pattern analysis and machine intelligence*, PAMI-9(5):698–700.
- [2] Blanco, J.-L. (2010). A tutorial on se (3) transformation parameterizations and on-manifold optimization. *University of Malaga, Tech. Rep.*
- [3] Brookshire, J. and Teller, S. (2013). Extrinsic Calibration from Per-Sensor Egomotion. In *Robotics: Science and Systems VIII*, pages 504–512. MIT Press.
- [4] Endres, F., Hess, J., Sturm, J., Cremers, D., and Burgard, W. (2014). 3-D mapping with an RGB-D camera. *IEEE Transactions on Robotics*, 30(1):177–187.
- [5] Fankhauser, P., Bloesch, M., Rodriguez, D., Kaestner, R., Hutter, M., and Siegwart, R. (2015). Kinect v2 for mobile robot navigation: Evaluation and modeling. In *Advanced Robotics (ICAR), 2015 International Conference on*, pages 388–394. IEEE.
- [6] Fernández-Moral, E., Gonzalez-Jimenez, J., and Arévalo, V. (2015). Extrinsic Calibration of 2D Laser Rangefinders from Perpendicular Plane Observations. *International Journal of Robotics Research*, online.
- [7] Fernandez-Moral, E., Gonzalez-Jimenez, J., Rives, P., and Arevalo, V. (2014). Extrinsic calibration of a set of range cameras in 5 seconds without pattern. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 429–435. IEEE.
- [8] Gomez-Ojeda, R., Briaies, J., Fernandez-Moral, E., and Gonzalez-Jimenez, J. (2015). Extrinsic calibration of a 2d laser-rangefinder and a camera based on scene corners. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 3611–3616. IEEE.
- [9] Gomez-Ojeda, R. and Gonzalez-Jimenez, J. (2016). Robust stereo visual odometry through a probabilistic combination of points and line segments. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2521–2526.
- [10] Ha, J.-E. (2012). Extrinsic calibration of a camera and laser range finder using a new calibration structure of a plane with a triangular hole. *International Journal of Control, Automation and Systems*, 10(6):1240–1244.

- [11] Moré, J. J. (1978). The Levenberg-Marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer.
- [12] Moreno, F.-A., Blanco, J.-L., and González-Jiménez, J. (2013). ERODE: An efficient and robust outlier detector and its application to stereovisual odometry. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4691–4697. IEEE.
- [13] Nex, F. and Remondino, F. (2014). UAV for 3D mapping applications: a review. *Applied geomatics*, 6(1):1–15.
- [14] Nguyen, C. V., Izadi, S., and Lovell, D. (2012). Modeling kinect sensor noise for improved 3d reconstruction and tracking. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 524–530. IEEE.
- [15] Schneider, S., Luettel, T., and Wuensche, H. J. (2013). Odometry-based online extrinsic sensor calibration. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1287–1292.
- [16] Sorkine, O. (2009). Least-squares rigid motion using SVD. *Technical notes*, 120(3):52.
- [17] Staranowicz, A., Brown, G. R., Morbidi, F., and Mariottini, G. L. (2014). *Easy-to-Use and Accurate Calibration of RGB-D Cameras from Spheres*, pages 265–278. Springer Berlin Heidelberg.
- [18] Svoboda, T., Martinec, D., and Pajdla, T. (2005). A convenient multicamera self-calibration for virtual environments. *PRESENCE: teleoperators and virtual environments*, 14(4):407–422.
- [19] Taylor, Z. and Nieto, J. (2015). Motion-based calibration of multimodal sensor arrays. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4843–4850.