MÉTODO DE ODOMETRÍA BASADA EN PLANOS PARA CÁMARAS DE PROFUNDIDAD

Andres Galeote-Luque, Jose-Raul Ruiz-Sarmiento, Javier Gonzalez-Jimenez
Machine Perception and Intelligent Robotics Group,
Universidad de Málaga, Campus de Teatinos, 29071, Málaga, España
{andresgalu, jotaraul, javiergonzalez}@uma.es

Resumen

En este artículo se presenta un método para el cálulo de la odometría en tiempo real de una cámara de profundidad, tipicamente una cámara RGB-D, donde solo se emplea la imagen D. El método propuesto tiene la ventaja frente a las alternativas existentes en el estado del arte de ser eficiente a la vez que ofrece una precisión superior. Para ello, la estimación del movimiento entre dos imágenes de entrada se calcula minimizando la distancia entre trozos planos de una imagen y puntos de la otra. La propuesta incluye un procedimiento iterativo que refina la solución (trasformación rígida entre dos imágenes consecutivas) mediante la actualización de los emparejamientos de pares de puntos y planos característicos hasta la convergencia. El método presentado se ha evaluado y comparado con una técnica del estado del arte, mostrando una reducción del 25% de la mediana de los errores de translación y rotación, funcionando a la misma frecuencia (~30Hz).

Palabras clave: Odometría visual, cámaras RGB-D.

1 INTRODUCCIÓN

La posibilidad de calcular la trayectoria seguida por una cámara, problema conocido como *Odometría Visual* (en inglés *Visual Odometry*, VO) [11], es de relevancia en multitud de campos, como la robótica movil, coches autonomos, UAVs, realidad aumentada, etc. Aunque tradicionalmente el término VO ha sido asociado al uso de cámaras de color o intensidad, en la actualidad también se incluyen a las cámaras de profundidad o RGB-D. La introducción de estas cámaras ha propiciado el desarrollo de nuevos métodos de VO que aprovechan la información geométrica de la escena que proporcionan.

Los campos anteriormente mencionados requieren de métodos de VO rápidos y precisos, con el fin de poder ejecutarse en disposivos con recursos computacionales limitados o compartidos, razón por la que los métodos conocidos como *directos* se imponen en la literatura sobre VO aplicada a cámaras RGB-D. Dentro de esta categoría de métodos, destaca

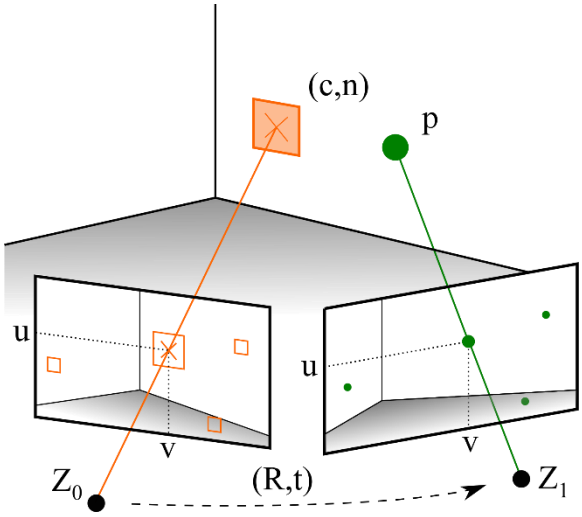

Figura 1: Dos imágenes consecutivas de una cámara de profundidad (RGB-D) y los elementos característicos empleados por el método presentado. Los planos [centro (**c**), vector normal (**n**)] se extraen de la imagen de profundidad Z_0, y son emparejados con los puntos **p** de la imagen Z_1 localizados en las mismas coordenadas de la imagen. La estimación del movimiento se obtiene minimizando la distancia entre los pares <punto, plano>, refinando el resultado posteriormente mediante un proceso iterativo en el que se actualizan los puntos **p** seleccionados.

DIFODO [8], dado que supera en precisión y rapidez a métodos similares aun haciendo uso únicamente de la información geométrica, descartando la imagen radiométrica. Las virtudes que presenta este método nacen de su novedosa formulación, que sustituye los tipicos emparejamientos entre caracerisiticas de las imágenes de los métodos indirectos por una restricción sobre el movimiento de los puntos en 3D, que se conoce como restricción del flujo de rango [8, 6]. El método DIFODO, especifico para cámaras de profundidad, aprovecha la información geométrica de toda la imagen para reducir el efecto del ruido de los datos y de las desviaciones sobre esta restricción, que aplica solo a puntos en superficies planas. Además, con el fin de aumentar la precisión y tolerancia a grandes movimientos, aplica un proceso iterativo en el que una de las imágenes es reproyectada sobre la otra en función de la estimación de la pose más reciente. A

pesar de las ventajas mencionadas, esta formulación también presenta ciertas limitaciones, ya que se parte de la hipótesis de que el entorno de cada píxel es aproximable por un plano. Al incluir todos los píxeles, aquellos que no cumplan esta hipótesis pueden repercutir negativamente en la estimación del movimiento. Por otro lado, la reproyección de las imágenes es un proceso no lineal y computacionalmente costoso, propenso a crear artefactos cerca de oclusiones en la imagen reproyectada.

En este artículo presentamos un método de odometría visual capaz de trabajar en tiempo real con imágenes de profundidad de cámaras RGB-D manteniendo una alta precisión, y que aborda las limitaciones mencionadas anteriormente. La figura 1 muestra, de manera simplificada, el funcionamiento del método. En resumen, se seleccionan y extraen regiones planas características (*planar features*) y puntos característicos (*keypoints*) de la primera y segunda imagen respectivamente, utilizados posteriormente para obtener la estimación de la pose minimizando la distancia entre pares <punto, plano>. El objetivo de realizar esta selección de elementos característicos es centrar los recursos computacionales en la información encontrada en un conjunto de regiones de la imagen, seleccionadas de forma que se optimice dicha información. Para este fin, el método presentado calcula la planicidad del entorno de cada píxel de la imagen de profundidad analizada, con el objetivo de seleccionar los píxeles que representen superficies planas de la escena. Este proceso se realiza de forma rápida y eficiente mediante convoluciones 2D. A pesar del uso de puntos y regiones planas características, no se sigue un procedimiento tradicional de emparejamiento de correspondencias, y en su lugar se calculan los emparejamientos entre elementos característicos de forma iterativa en función de la estimación más reciente de la pose relativa. Este emparejamiento iterativo es similar al realizado en DIFODO, con la diferencia de que solo se reproyectan los elementos característicos seleccionados en vez de la totalidad de la imagen, reduciendo así el coste computacional de la operación. Cabe destacar que, al minimizar distancias entre puntos y planos, el emparejamiento entre dos elementos carácterísticos es válido si ambos representan la misma superficie plana de la escena, por lo que se admite cierto margen de error.

Las diferencias que el método propuesto presenta frente a DIFODO crean por otro lado similitudes con el reconocido algoritmo ICP [2] (del inglés *Iterative Closest Point*), y en particular con sus variantes que se basan en el uso de la distancia entre puntos y planos [4, 14]. Sin embargo estos métodos proporcionan la pose relativa entre dos nubes de puntos desordenadas, mientras que en este método se aprovecha la representación de la escena en forma de imagen de profundidad. Para empezar, la representación ordenada de la información geométrica permite localizar los puntos que componen el entorno de un píxel sin necesidad de realizar un algoritmo de búsqueda costoso. Además, la función de proyección de la cámara de profundidad se utiliza para emparejar las regiones características obtenidas de la primera imagen con los puntos característicos de la segunda.

Con el fin de validar el método presentado, se ha comparado con el método DIFODO [8], empleando como banco de pruebas el conjunto de datos ICL-NUIM RGB-D [7]. Los resultados muestran como nuestro método es capaz de estimar la pose de la cámara con mayor precisión, reduciendo un 25% la mediana del error en translación y rotación, manteniendo una velocidad superior a la frecuencia de trabajo del sensor.

2 TRABAJOS RELACIONADOS

La aparición de cámaras RGB-D hace más de una decada supuso la creación de un nuevo campo dentro de la odometría visual que ha sido extensamente estudiado a lo largo de los años. Los métodos desarrollados para calcular la VO haciendo uso de cámaras RGB-D se dividen en dos grandes grupos, de forma similar a la odometría visual tradicional. Por una parte, los métodos indirectos se caracterizan por estimar la pose en función de un conjunto de elementos característicos (en ingles *features*) extraídos de las imágenes. Por el contrario, los métodos directos optimizan una función de costes relacionada con la diferencia entre las imágenes analizadas.

Los métodos indirectos son los más comunes en la literatura, entre los cuales se pueden encontrar enfoques que usan diferentes tipos de elementos característicos, como puntos [12], una combinación de líneas y puntos [10], y planos [13, 3]. El correcto funcionamiento de los métodos indirectos depende en gran medida de la correcta selección y emparejamiento de elementos característicos, por lo que una gran parte del coste computacional se emplea en este apartado. En el caso de usar planos característicos, se acostumbra a realizar una segmentación de planos, cuya complejidad puede aumentar considerablemente el tiempo de ejecución.

Por el contrario, los métodos directos evitan el preprocesado necesario para obtener los elementos característicos al trabajar directamente con las imágenes. En [15] se presenta este concepto, en el que se minimiza el error relacionado con la información radiométrica mientras que la información geométrica se usa únicamente para reproyectar las imágenes. Este

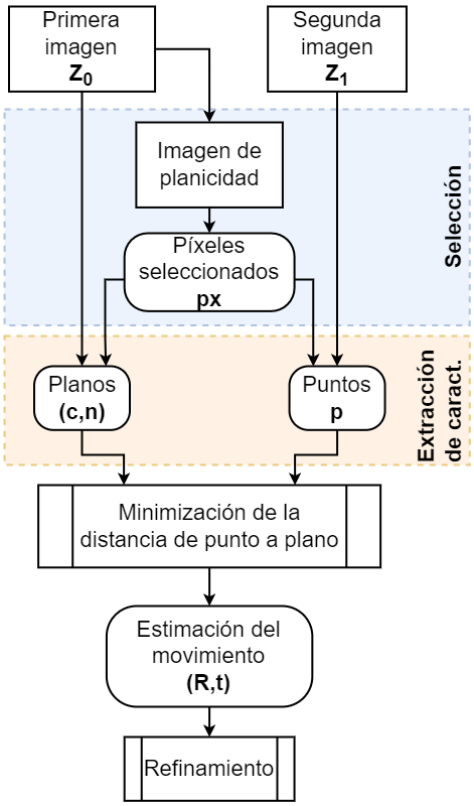

Figura 2: Diagrama del funcionamiento del método presentado. La entrada es un par de imágenes de profundidad. De la primera se obtiene una imagen de planicidad, usada para seleccionar los píxeles ubicados en zonas planas de la escena. De las imágenes se extraen regiones planas (Z_0) y puntos característicos (Z_1). Minimizando la distancia entre ellos se obtiene la estimación del movimiento. Por último, se aplica un proceso iterativo para refinar la solución (pose relativa entre ambas).

trabajo se extendió en [9] introduciendo una formulación probabilística que aumenta la robustez. Por otro lado en [8] se ignora la imagen de color, y se usa la restricción de flujo de rango en toda la imagen de profundidad. El resultado es un método muy rápido capaz de conseguir una precisión mayor que otros métodos similares.

A pesar de no ser considerado un método de odometría, el algoritmo ICP [2] y sus variantes han sido utilizados en la literatura como punto de comparación de métodos de VO con cámaras RGB-D. El objetivo de ICP es el registro de dos nubes de puntos desordenadas, y dado que las imágenes de profundidad son una representación ordenada de una nube de puntos, se puede aplicar ICP para obtener la pose relativa del sensor dado un par de imágenes. A lo largo de los años, se han desarrollado diferentes versiones del algoritmo ICP: en [4] se minimiza la distancia entre puntos y planos en vez de la distancia entre pares de puntos como en el algoritmo original; en [14] se combinan los dos métodos mencionados previamente en una única formulación probabilística. No obstante, el algoritmo y sus diferentes versiones suelen resultar computacionalmente costosas al requerir el cálculo de distancias y emparejamientos entre un elevado número de puntos/características.

El método presentado en este artículo persigue ser rápido a la vez que preciso, para lo que se parte de una idea similar a la introducida por DIFODO, y se palian sus limitaciones mediante la combinación de conceptos usados en diferentes técnicas. Al hacer uso de elementos característicos se centran los recursos computacionales en las partes de la imagen que representan superficies planas de la imagen. Sin embargo, tampoco se hace un emparejamiento tradicional de los elementos característicos, y se opta en su lugar por actualizar de forma iterativa el emparejamiento de los puntos y planos característicos en función de la estimación de la pose.

3 ODOMETRÍA PARA CÁMARAS RGB-D

En esta sección se resume cómo el método propuesto estima el movimiento de la cámara RGB-D a lo largo de una secuencia de imágenes de profundidad. El objetivo es, por lo tanto, obtener la pose relativa o trasformación rígida del sensor entre dos imágenes consecutivas, definida por una matriz de rotación y un vector de traslación $(R, \mathbf{t})$, siendo $R \in \mathrm{SO}(3)$ y $\mathbf{t} \in \mathbb{R}^3$. La información de entrada es un par de imágenes de profundidad referidas como Z_0 y Z_1, siendo $Z_i: \Omega \rightarrow \mathbb{R}$ definida en el plano imagen $\Omega \subset \mathbb{R}$ (nótese que Z_0 precede temporalmente a Z_1). La figura 2 muestra un diagrama del flujo de trabajo del método presentado.

La estimación de la pose se calcula minimizando la distancia entre pares de elementos característicos, compuestas por regiones planas extraídas de Z_0 y puntos de Z_1. Para que la solución sea válida, ambos elementos deben pertenecer a la misma superficie plana de la escena, por lo que una correcta selección es necesaria para aumentar la probabilidad de que se cumpla la hipótesis. Con este fin, se crea una imagen que representa la planicidad del vecindario de cada píxel, que es posteriormente empleada para seleccionar los píxeles que pertenecen a superficies planas de la escena. Este procedimiento de selección será explicado con más detalle en la sección 3.1.

Tras la selección, se procede a ajustar un plano al vecindario de cada píxel seleccionado en Z_0, definido por su centro y vector normal $(\mathbf{c_i}, \mathbf{n_i})$. Cada punto correspondiente $\mathbf{p_i}$ se extrae de Z_1 en función de la pose relativa inicial entre ambas imágenes. En caso de no tener una noción previa sobre la pose relativa, se asume que los puntos seleccionados se localizan en las

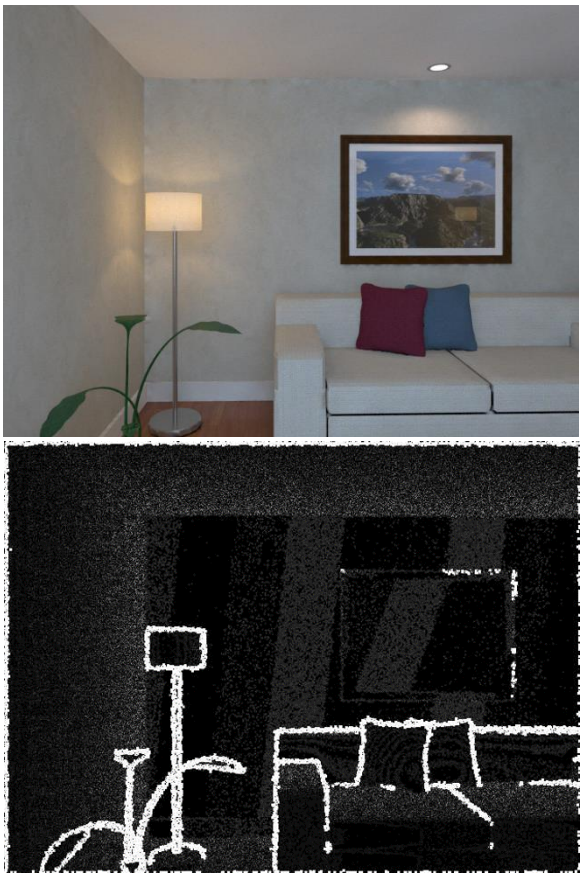

Figura 3: Imagen de una escena (arriba) y su correspondiente imagen de planicidad (abajo). Los píxeles claros representan zonas en las que el entorno del píxel no puede ser representado por un plano, como los bordes de los objetos.

mismas coordenadas de la imagen que las regiones planas seleccionadas. En el caso contrario, se puede aplicar el flujo óptico a los centros de los planos seleccionados para reproyectarlos sobre la imagen Z_1, encontrando así los píxeles de los que extraer los puntos correspondientes. En la sección 3.2 se analizará el proceso de extracción de características en función de la pose relativa.

Una vez se ha seleccionado y extraído de Z_0 el conjunto de planos (C, N), siendo $\mathbf{c_i} \in C$ y $\mathbf{n_i} \in N$, y el conjunto de puntos P de Z_1, con $\mathbf{p_i} \in P$, el siguiente paso consiste en encontrar la transformación relativa $(R, \mathbf{t})$ que minimice la distancia entre los pares de características. La estimación del movimiento será válida si se cumple la hipótesis de que cada pareja de elementos característicos representa la misma superficie plana de la escena. La estimación de la pose será detallada en la sección 3.3.

Tras obtener la primera estimación del movimiento, se aplica un proceso iterativo de refinado en el que el emparejamiento entre los elementos característicos se recalcula en cada iteración en función de la pose relativa más reciente, haciendo más probable que los pares de puntos y planos característicos se encuentren en la misma superficie de la escena. La estimación de la pose se vuelve a obtener usando los nuevos pares <punto, plano>, y se repite hasta que la solución converja. En la sección 3.4 se explicarán más a fondo las fases del refinado.

Por último, en la sección 3.5 se analizarán las limitaciones del método presentado, principalmente la necesidad de contar con planos en tres direcciones ortogonales para obtener una solución válida. También se explicará el paso de filtrado de la solución aplicado para paliar los efectos de dichas limitaciones.

3.1 SELECCIÓN Y EXTRACCIÓN DE PLANOS

En esta sección se detalla el procedimiento de selección y extracción de planos, con el fin de reducir la imagen de entrada Z_0 a un conjunto de planos definidos por su centro C y vector normal N, y Z_1 a un conjunto de puntos P que hipotéticamente pertenezcan a dichos planos.

Con el propósito de maximizar las posibilidades de que un punto $\mathbf{p_i}$ pertenezca al plano correspondiente $(\mathbf{c_i}, \mathbf{n_i})$ los planos se extraen de ciertos píxeles seleccionados según la planicidad de su entorno. Cuanto mayor sea la superficie plana en la que se encuentra el píxel seleccionado, mayor será la probabilidad de que dicho píxel apunte a la misma superficie de la escena en ambas imágenes, cumpliendo por lo tanto la hipótesis previamente mencionada. Podría emplearse un algoritmo de segmentación de planos para seleccionar píxeles que representen superficies planas, sin embargo estos métodos suelen ser computacionalmente costosos y lentos.

En el método presentado se hace uso de convoluciones 2D para obtener una imagen de planicidad que da información de cómo de plano es el entorno de cada píxel. Esto es posible ya que el vecindario de un píxel de la imagen de profundidad presenta simetría cuando este representa un punto con un entorno plano. Midiendo la diferencia entre la profundidad del píxel central y la media de la profundidad de su entorno se puede obtener una buena aproximación de la planicidad de dicho entorno. Como se ha mencionado previamente, esta operación se puede implementar de manera rápida y eficiente mediante la realización de una convolución 2D sobre la totalidad de la imagen, usando el kernel laplaciano mostrado en la ecuación (1). En la figura 3 se muestra una imagen de color de la escena, acompañada de la imagen de planicidad correspondiente.

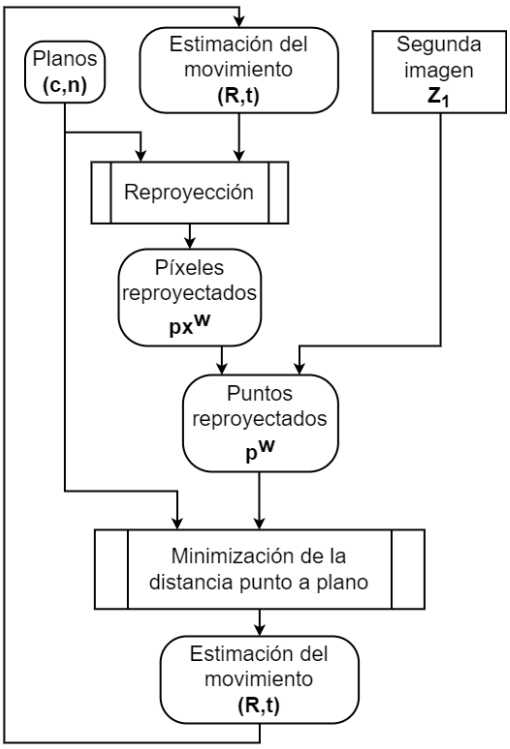

Figura 4: Diagrama del proceso iterativo de refinado aplicado a la solución. La estimación de la pose se utiliza para reproyectar los centros de los planos sobre la imagen Z_1, de la que se obtienen los puntos en las nuevas coordenadas. La estimación del movimiento se puede recalcular minimizando la distancia entre los planos y los puntos actualizados.

$$\frac{1}{8}\begin{bmatrix}1 & 1 & 1\\ 1 & -8 & 1\\ 1 & 1 & 1\end{bmatrix} \qquad (1)$$

Una vez se ha creado la imagen de planicidad a partir de Z_0 se procede a la selección de los píxeles localizados en los entornos más planos. Para asegurar una correcta distribución de píxeles seleccionados por toda la imagen, esta se divide en bloques, de tal manera que se seleccionan aquellos píxeles de cada bloque con un mayor índice de planicidad.

El siguiente paso consiste en ajustar un plano $(\mathbf{c_i}, \mathbf{n_i})$ a cada uno de los píxeles seleccionados en la imagen Z_0. El centro del plano es simplemente el centroide del vecindario del píxel, y el vector normal se obtiene mediante la aplicación de la Descomposición en Valores Singulares (DVS) a la matriz de diferencias entre cada punto del entorno y el centroide. El vector normal $\mathbf{n_i}$ es por lo tanto el vector singular izquierdo asociado al menor valor singular. Este valor singular puede además ser usado como una medida de la calidad del ajuste del plano al entorno, o *fitness* f_i, de forma que un menor *fitness* representa un mejor ajuste.

Es importante señalar que aunque el uso exclusivo de la imagen de planicidad no provee garantía de que los píxeles seleccionados estén ubicados en superficies planas de la escena, su *fitness* puede ser usado en pasos posteriores para reducir el peso de las características extraídas de superficies curvas o cercanas a oclusiones.

3.2 EXTRACCIÓN DE PUNTOS

En esta sección se explica la extracción de puntos de la imagen Z_1 basado en la pose relativa inicial entre ambas imágenes. Al trabajar con imágenes consecutivas, y tras seleccionar los píxeles localizados en superficies planas de la escena, se pueden emparejar los planos seleccionados de Z_0 con los puntos obtenidos de Z_1 ubicados en las mismas coordenadas del plano imagen. La hipótesis a cumplir para que este emparejamiento sea válido es que las parejas de puntos y planos característicos seleccionados pertenezcan a la misma superficie de la escena, por lo que no es necesario un emparejamiento perfecto entre los elementos característicos. Cabe destacar el uso de un filtro gaussiano al entorno de cada uno de los píxeles seleccionados para reducir el efecto del ruido de la imagen.

En el caso de tener una noción previa de la estimación de la pose, esta puede ser usada para mejorar el emparejamiento entre los pares <punto, plano> mediante el flujo óptico. De ser así, el procedimiento a realizar sería el siguiente:

1. El centro de cada plano seleccionado $\mathbf{c_i}$ se transforma usando la estimación del movimiento $(R, \mathbf{t})$ para obtener $\mathbf{c_i^w}$.

$$\mathbf{c_i^w} = [x_c^w, y_c^w, z_c^w]^T = R^{-1}(\mathbf{c_i} - \mathbf{t}) \qquad (2)$$

2. Se aplica la función de proyección de la cámara RGB-D para obtener las coordenadas (v, u) en el plano imagen del centro transformado.

$$v = f_y \frac{y_c^w}{z_c^w} + v_m$$
$$u = f_x \frac{x_c^w}{z_c^w} + u_m \qquad (3)$$

3. Estas nuevas coordenadas se pueden usar para extraer el nuevo punto $\mathbf{p_i^w}$, aplicando un filtro gaussiano como se ha comentado previamente.

3.3 ESTIMACIÓN DE LA POSE

En esta sección se explora la manera de estimar la pose relativa que minimice la distancia de punto a plano de

cada pareja de elementos característicos. Como se ha mencionado previamente, se hace uso del valor del *fitness* de cada plano para quitar peso a aquellas regiones cuyo entorno tenga una baja planicidad. Se incluye además un peso en función de la profundidad del centro, dado que las cámaras RGB-D presentan un ruido proporcional a la distancia del objeto al sensor [17]. El peso combinado es por lo tanto calculado como $\alpha_i = (1 - f_i) + z_c^2/25$. La función de pérdida de Huber ρ mostrada en (4) se aplica al residual con el fin de aumentar la robustez frente a *outliers*, es decir, parejas de características que no cumplan las hipótesis mencionadas previamente. Esta resulta:

$$\rho(e) = \begin{cases} {}^1/_2\, e^2 & si\ |e| \le k \\ k|e| - {}^1/_2\, k^2 & si\ |e| > k \end{cases} \qquad (4)$$

La ecuación (5) representa la minimización de la distancia entre los pares <punto, plano> tras aplicar el peso α y la función de pérdida de Huber, y se optimiza mediante la implementación de Levenberg-Marquardt de Ceres [1].

$$(R, \mathbf{t}) = \arg\min_{R,\boldsymbol{t}} \sum_i \rho(\|\alpha_i \mathbf{n_i} \cdot (R\mathbf{p_i} + \mathbf{t} - \mathbf{c_i})\|^2) \qquad (5)$$

3.4 REFINADO DE LA SOLUCIÓN

Una vez obtenida la primera estimación de la pose relativa, se refina el resultado mediante un proceso iterativo en el que se recalcula el emparejamiento entre los elementos característicos para así obtener una mejor estimación del movimiento. La figura 4 muestra un esquema de las fases del refinamiento. El emparejamiento se actualiza en función de la estimación de la pose haciendo uso del flujo óptico tal y como se ha explicado en la sección 3.2. En resumen, se reproyectan los centros de los planos C sobre la imagen Z_1 en función de la transformación $(R, \mathbf{t})$, y las coordenadas resultantes se usan para extraer nuevos puntos P^{w}. Usando los emparejamientos actualizados se puede recalcular la estimación de la pose minimizando la distancia entre ellos, aplicando la ecuación (5) pero con los nuevos puntos $\mathbf{p_i^w} \in P^w$. Este proceso se repite iterativamente hasta que la solución converja.

3.5 FILTRADO DE LA SOLUCIÓN

El uso de las distancias de punto a plano conlleva la ventaja, como se ha mencionado previamente, de permitir un emparejamiento de elementos característicos más laxo. Sin embargo, la posición de cada punto dentro de su correspondiente plano no queda restringida, por lo que es necesario obtener un conjunto de planos ortogonales para recuperar el movimiento de manera fiable. Un ejemplo de escena en la que no hay información suficiente para recuperar el movimiento es un pasillo, ya que el movimiento del sensor a lo largo del pasillo no está restringido. Para paliar esta limitación se ha implementado un filtro, inspirado en el empleado en la técnica DIFODO [8], que combina la estimación del movimiento actual con la calculada para el par de imágenes anteriores en caso de que la escena no se encuentre restringida en todas las dimensiones.

Tabla 1: RMSE del error de translación.

	(cm/frame)		(cm/s)	
	Difodo	Propuesta	Difodo	Propuesta
Sec. 0	**0.4106**	0.4211	**9.9206**	10.1152
Sec. 1	0.2259	**0.1823**	4.9823	**4.1853**
Sec. 2	**0.4292**	0.7793	**8.9030**	21.2612
Sec. 3	0.6851	**0.4588**	15.9127	**11.3200**

Tabla 2: RMSE del error de rotación.

	(º/frame)		(º/s)	
	Difodo	Propuesta	Difodo	Propuesta
Sec. 0	0.2278	**0.2248**	5.4471	**3.5111**
Sec. 1	0.0518	**0.0471**	0.7426	**0.6341**
Sec. 2	0.0815	**0.0682**	1.3769	**1.3029**
Sec. 3	0.1861	**0.1262**	3.2798	**1.5959**

4 EXPERIMENTOS Y RESULTADOS

En esta sección se evalúa el desempeño del método presentado sobre el dataset ICL-NUIM RGB-D [7] y se comparan los resultados con los obtenidos por DIFODO [8]. Para todos los experimentos se ha usado un PC con 8GB de RAM a 2400MHz, Ubuntu 20.04 y una CPU Intel Core i7-7700HQ a 2.8GHz.

4.1 DATASET

La evaluación del método se realiza sobre las 4 secuencias de interior denominadas *Living Room* del dataset virtual ICL-NUIM RGB-D. Este banco de pruebas destaca por el uso de trayectorias realistas empleadas dentro de un entorno simulado, lo que permite conocer sin error la pose de la cámara en todo momento. Incluye además el ruido típicamente presente en las imágenes de profundidad al usar cámaras RGB-D. Todas las secuencias han sido grabadas a una frecuencia de 30Hz con una resolución de 640x480.

4.2 RESULTADOS

Para comparar los resultados del método presentado con DIFODO [8] se emplea la métrica presentada en [16], conocida como RPE (del inglés *Relative Pose*

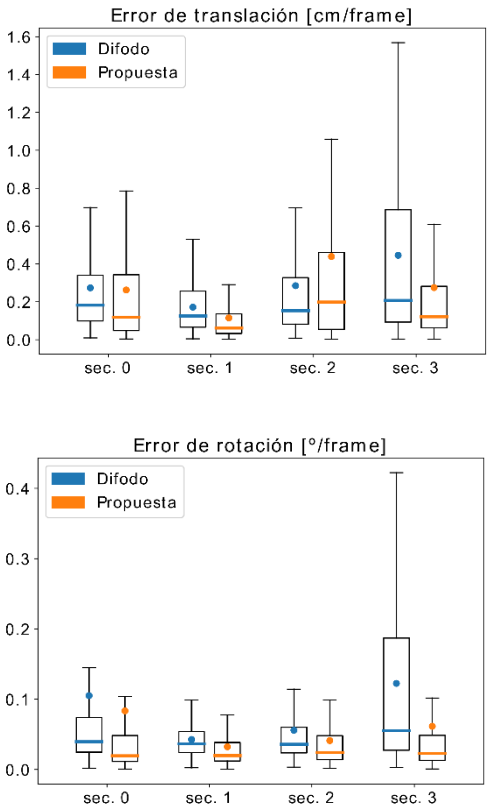

Figura 5: Resultados de ambos métodos en las secuencias *Living Room* del dataset ICL-NUIM RGB-D. Las cajas representan los cuartiles centrales, con la línea coloreada siendo la mediana. La media de los datos se muestra como un punto coloreado.

Error). En cuanto al tiempo de ejecución en ambos casos es de ~30ms por *frame*, por lo que son capaces de funcionar a tiempo real ya que la frecuencia de refresco de este tipo de sensores suele ser de 30Hz.

En la figura 5 se muestran los diagramas de cajas de los resultados en translación (arriba) y rotación (abajo). Se puede observar como la mediana del error de translación de el método propuesto es menor que la de DIFODO en todas las secuencias excepto la número 2, mientras que en rotación sus resultados son siempre más precisos. De media, el método presentado reduce la mediana de los errores de translación y rotación por *frame* en un 25%, y por segundo en un 40%. En las tablas 1 y 2 se muestra el RMSE de los errores en las diferentes secuencias. En translación, ambos métodos consiguen un nivel similar de precisión, mientras que en rotación conseguimos reducir el error en un 14%.

La reducción del error obtenida de los resultados se traduce en una mejor localización del sensor a lo largo de la secuencia, ya que el error cometido al calcular la pose relativa entre un par de imágenes se acumula, lo que se conoce como *drift*.

5 CONCLUSIONES

En este artículo se ha presentado un nuevo método para la obtención rápida y precisa de odometría visual de cámaras RGB-D que consigue mitigar las limitaciones de los métodos directos al aplicar un proceso de selección de elementos característicos. Este método hace uso de planos y puntos característicos para estimar la transformación rígida que minimice la distancia entre pares <punto, plano>. De esta forma se aprovecha la información proporcionada por las superficies planas de la escena que están comúnmente presentes en entornos artificiales de interior. Con el fin de refinar la estimación del movimiento se incluye un procedimiento iterativo mediante el cual se actualizan los emparejamientos de los elementos característicos en función de la pose relativa hasta la convergencia.

Para comprobar la viabilidad del método propuesto se ha comparado con DIFODO [8], método directo conocido por su precisión y rapidez. Los resultados muestran como nuestro método es capaz de reducir un 25% la mediana del error en translación y rotación.

English summary

ODOMETRY METHOD BASED ON PLANES FOR DEPTH CAMERAS

Abstract

In this article a novel method is presented that computes the odometry of a depth camera (RGB-D) in real time, using only the depth information. The proposed method offers efficiency and higher accuracy than other alternatives found in the literature. For that end, the motion is recovered by minimizing the point-to-plane distance between pairs of features, leveraging the information provided by the flat surfaces of the scene typically found in manmade environments. The proposal includes an iterative approach used to refine the solution (rigid transformation between two consecutive images) by updating the matching between the features until convergence. The method has been tested and compared with a state-of-the-art method, resulting in a reduction of 25% of the median of the translational and rotational error, while working at the same frequency (~30Hz).

Keywords: Visual odometry, RGB-D cameras.

Referencias

[1] Agarwal, S., & Mierle, K. (2012). Ceres solver, http://ceres-solver.org.

[2] Besl, P. J., & McKay, N. D. (1992, April). Method for registration of 3-D shapes. In *Sensor fusion IV: control paradigms and data structures* (Vol. 1611, pp. 586-606). Spie.

[3] Chen, B., Liu, C., Tong, Y., & Wu, Q. (2017, July). Robust RGB-D Visual Odometry Based on Planar Features. In *2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)* (pp. 449-453). IEEE.

[4] Chen, Y., & Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and vision computing*, 10(3), 145-155.

[5] Engel, J., Koltun, V., & Cremers, D. (2017). Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, *40*(3), 611-625.

[6] Gonzalez, J., & Gutierrez, R. (1999). Direct motion estimation from a range scan sequence. *Journal of Robotic Systems*, 16(2), 73-80.

[7] Handa, A., Whelan, T., McDonald, J., & Davison, A. J. (2014, May). A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *2014 IEEE international conference on Robotics and automation (ICRA)* (pp. 1524-1531). IEEE.

[8] Jaimez, M., & Gonzalez-Jimenez, J. (2015). Fast visual odometry for 3-D range sensors. *IEEE Transactions on Robotics*, *31*(4), 809-822.

[9] Kerl, C., Sturm, J., & Cremers, D. (2013, May). Robust odometry estimation for RGB-D cameras. In 2013 *IEEE international conference on robotics and automation* (pp. 3748-3754). IEEE.

[10] Lu, Y., & Song, D. (2015). Robust RGB-D odometry using point and line features. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 3934-3942).

[11] Nistér, D., Naroditsky, O., & Bergen, J. (2004, June). Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* (Vol. 1, pp. I-I). Ieee.

[12] Prakhya, S. M., Bingbing, L., Weisi, L., & Qayyum, U. (2015, May). Sparse depth odometry: 3D keypoint based pose estimation from dense depth data. In *2015 IEEE international conference on robotics and automation (ICRA)* (pp. 4216-4223). IEEE.

[13] Raposo, C., Lourenço, M., Antunes, M., & Barreto, J. P. (2013, September). Plane-based Odometry using an RGB-D Camera. In *BMVC* (Vol. 2, No. 5, p. 6).

[14] Segal, A., Haehnel, D., & Thrun, S. (2009, June). Generalized-icp. In *Robotics: science and systems* (Vol. 2, No. 4, p. 435).

[15] Steinbrücker, F., Sturm, J., & Cremers, D. (2011, November). Real-time visual odometry from dense RGB-D images. In *2011 IEEE international conference on computer vision workshops (ICCV Workshops)* (pp. 719-722). IEEE.

[16] Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012, October). A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems* (pp. 573-580). IEEE.

[17] Zuñiga-Noël, D., Ruiz-Sarmiento, J. R., & Gonzalez-Jimenez, J. (2019, September). Intrinsic calibration of depth cameras for mobile robots using a radial laser scanner. In *International Conference on Computer Analysis of Images and Patterns* (pp. 659-671). Springer, Cham.