Universitat
de les Illes Balears

Uso combinado de planos y contornos de objetos en un odómetro visual para entornos estructurados débilmente texturados

Joan P. Company-Corcoles, Alberto Ortiz

Departament
de Ciències Matemàtiques
i Informàtica

TECHNICAL REPORT
Noviembre / 2018
A-02-2018

Uso combinado de planos y contornos de objetos en un odómetro visual para entornos estructurados débilmente texturados

Joan P. Company-Corcoles y Alberto Ortiz
Departamento de Ciencias Matemáticas e Informática
Universidad de las Islas Baleares
07122 Palma de Mallorca, España
joanpep.company@uib.es, alberto.ortiz@uib.es

***Resumen*—Aunque los métodos de odometría visual han evolucionado enormemente en los últimos años, aún existen entornos en los que este tipo de algoritmos no funcionan correctamente. Estos entornos se caracterizan por la ausencia de textura y la abundancia de geometrías planas. En este artículo, se presenta un odómetro visual que, mediante la utilización de un sensor RGB-D y la combinación de contornos y planos, es capaz de solventar los problemas que típicamente aparecen en los entornos estructurados. El método propuesto consta de dos etapas: en la primera se estima el movimiento entre imágenes mediante los contornos de los objetos de la escena y se detectan imágenes clave; en la segunda, para cada imagen clave generada, se detectan y extraen los planos de la escena, se emparejan los planos entre varias imágenes clave y se optimiza la transformación con una combinación de contornos y planos. Además de la odometría visual, presentamos un nuevo método de asociación de planos entre dos imágenes. Este método se basa en la evaluación de restricciones representadas sobre un grafo. También se compara el odómetro visual propuesto con otros odómetros y sistemas de localización y mapeo simultaneo mediante el banco de pruebas RGB-D TUM. En los resultados experimentales, se observa que el método propuesto obtiene mejores resultados en entornos estructurados, mientras que los resultados son muy similares en el resto de entornos.**

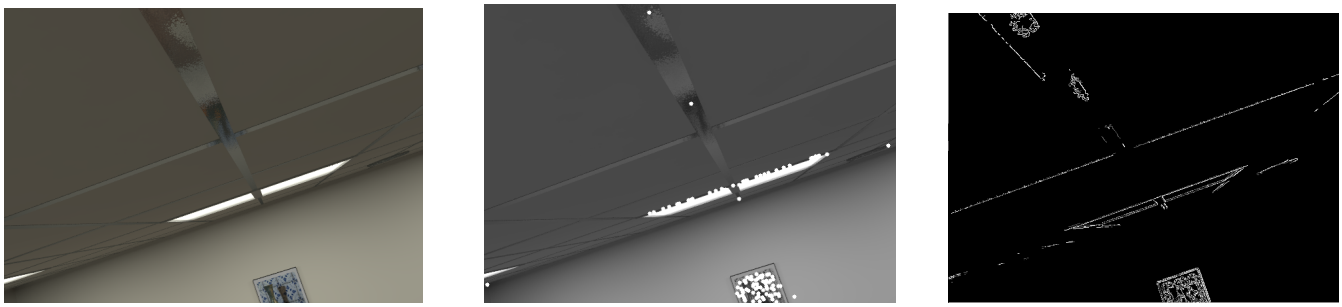
I. Introducción

La reconstrucción en tres dimensiones es un campo de aplicación muy importante tanto en el ámbito de la robótica como en el procesamiento de imágenes. En los últimos años, ha habido un gran avance en los algoritmos que calculan la posición de una cámara al mismo tiempo que realizan un mapa del entorno. Éstos son los conocidos métodos de SLAM (Simultaneous Localization and Mapping), los cuales típicamente comprenden un odómetro y una estratégia de detección de zonas previamente visitadas, situación que se conoce como cierre de bucle. La capacidad de detectar estas situaciones supone una ventaja crítica de los métodos SLAM, ya que permite relocalizar el sensor y corregir la deriva de la estimación. Otro de los puntos más importantes de las técnicas de SLAM es conseguir que el error de la trayectoria que se calcula entre imágenes sea pequeño antes de llegar a un cierre de bucle. Por esta razón, es necesario disponer de un odómetro lo más preciso posible.

Existen varios tipos de odómetros. En este artículo nos centraremos en los odómetros visuales. Se puede distinguir entre los que utilizan información de profundidad a nivel de píxel (usando cámaras RGB-D o estéreo) y las que no lo hacen (usan cámaras monoculares). Uno de los problemas de los sensores estereoscópicos que aún no pueden abordar es el cálculo de la profundidad en las escenas con falta de textura. La emergencia de sensores RGB-D, ligeros y económicos, ha solventado este problema. El cálculo de la profundidad en estos casos es posible gracias a la proyección de un patrón infrarrojo sobre la escena.

Aunque existen algoritmos de odometría y SLAM visual capaces de producir resultados precisos en muchos entornos, aún existen algunos retos que no son capaces de superar. Por ejemplo, la mayoría de ellos no son capaces de obtener buenos resultados en entornos estructurados y que presentan un gran cambio de luminosidad entre imágenes. Los odómetros visuales basados en puntos clave tienen problemas en entornos sin textura, donde el numero de puntos clave obtenidos es típicamente bajo o el emparejamiento de puntos entre imágenes resulta erróneo. A diferencia de los puntos clave, los métodos directos suelen fallar en la estimación de la posición de la cámara en situaciones donde el punto de vista o la intensidad lumínica cambian bruscamente. Otra alternativa para el registro de imágenes son los métodos de registro de puntos 3D mediante la evaluación de los puntos más cercanos en ICP (Iterative Closest Point). Sin embargo, ICP tiende a fallar si no se le proporciona una buena transformación inicial. En la figura 1, se ilustra otro problema de las soluciones estándar. En esta figura, se muestra un ejemplo de detección de puntos característicos utilizando el descriptor invariante a transformaciones de escala SIFT (Scale-Invariant Feature Transform) [1] y un detector de contornos Canny [2]. Se puede observar que en este caso es más útil la información de contornos que los puntos SIFT. Como es bien sabido, para poder realizar un registro lo más preciso posible es necesario disponer de puntos, cuantos más mejor, a lo largo de toda la imagen.

En este trabajo, presentamos un odómetro visual que combina planos y contornos, ambos predominantes en entornos estructurados. El objetivo de esta combinación es conseguir una mejora en la representación de las escenas, disminuir el error acumulado en la estimación de la posición de la cámara a lo largo del tiempo, así como minimizar los casos donde los otros sistemas tienden a fallar debido a insuficiencia de textura. Por ello, se ha optado por combinar información geométrica y fotométrica. Utilizando esta combinación se ha conseguido obtener resultados similares a los obtenidos por otros enfoques



(a) Imagen original (b) Puntos característicos SIFT (c) Contornos Canny

Figura 1: Detección de puntos característicos y contornos en una imagen de ejemplo de un entorno estructurado poco texturado

en entornos no estructurados y mejores resultados en entornos estructurados. Esta mejora se debe a la existencia de contornos, a diferencia de puntos clave, en lugares sin textura. Además, los contornos se obtienen a partir de información diferencial, lo que los hace tolerantes a cambios de iluminación entre imágenes. Por otra parte, el emparejamiento de planos permite alinear los contornos de forma precisa entre imágenes. Otra ventaja del método propuesto es que permite un registro más rápido y preciso que sólo utilizando puntos clave. Esto se debe a que el número de planos es bastante inferior al numero de puntos en una imagen, y que las variables que definen un plano están menos afectadas por el ruido de la cámara. Finalmente, la representación del mapa utilizando áreas planas ofrece varias ventajas en términos de almacenamiento y eficiencia computacional.

Las dos contribuciones de este trabajo son:

- Desarrollo de una técnica novedosa de emparejamiento de planos entre dos imágenes mediante restricciones propias del plano y relaciones entre los demás planos de la imagen utilizando un grafo.
- Desarrollo de un odómetro visual que combina contornos y planos.

Al final de este artículo se presenta una evaluación del odómetro propuesto comparándolo con métodos de odometría y SLAM visual en diferentes tipos de entornos mediante el banco de pruebas RGB-D TUM.

El resto del artículo se organiza de la siguiente forma: la sección II revisa los trabajos previos relacionados; la sección III describe la vista general del sistema; la sección IV incluye una técnica de emparejamiento de planos que utiliza información de apariencia y de relación geométrica entre planos; la sección V presenta un odómetro visual que combina contornos y planos; la sección VI reporta resultados experimentales en diferentes entornos así como de una comparación con otros odómetros y métodos de SLAM; y finalmente la sección VII concluye el artículo.

II. Trabajo previo relacionado

II-A. Identificación y emparejamiento de planos

Para poder utilizar entidades planas en el registro de imágenes es necesario emparejar planos entre imágenes para posteriormente calcular la transformación entre ellas. La mayor parte de las técnicas aplicadas se basan en utilizar la transformación entre imágenes calculada previamente mediante otras técnicas y utilizarla como entrada al sistema.

Uno de los métodos más utilizados consiste en obtener la transformación inicial entre imágenes mediante RANSAC (Random Sample Consensus) [3]. Para ello se eligen tres planos candidatos aleatorios, se estima una rotación y una traslación y se determina el número de muestras que corroboran esa transformación. El resultado se obtiene mediante un proceso iterativo que utiliza todas las combinaciones de tres planos y que elige la transformación que consigue explicar el mayor número de muestras [4]. Para simplificar este proceso iterativo, en [5], en lugar de elegir tres planos aleatorios, los candidatos son filtrados de acuerdo con su similitud a nivel de color y las relaciones angulares entre el plano evaluado y el resto de planos de la imagen. Además, este sistema utiliza RANSAC fusionando hipótesis de planos con un conjunto de puntos previamente emparejados. En [6] proponen un método de emparejamiento de planos utilizando restricciones geométricas en una versión modificada de RANSAC.

Otros enfoques obtienen la transformación entre imágenes en un proceso anterior al emparejamiento de planos. Por ejemplo, en [7], una vez ya se ha obtenido la transformación entre imágenes, la segunda imagen es proyectada sobre la imagen del instante anterior, y si el plano modelado interseca en el plano medido, entonces se establece una correspondencia. Otro método que sigue este principio se detalla en [8], donde, en lugar de la intersección de planos, se calcula un punto central para cada plano, y si este punto proyectado en la segunda imagen esta dentro de una cuadrícula, cuyo centro se corresponde con el centro del otro plano, entonces se establece una correspondencia. En [9], se basan en una técnica de solapamiento entre planos y una restricción geométrica que involucra sus ángulos y distancias. En [10], también proponen una restricción de solapamiento como técnica para asociar planos de diferentes imágenes.

Existen otros métodos que no dependen de una alineación inicial de imágenes. Este es el caso de [11], donde proponen un método en el cual para cada plano se genera un histograma gracias a una red neuronal, y a través de la evaluación de su distancia entre histogramas se realiza la identificación de planos. El sistema más parecido al nuestro es detallado en [12], donde describen un sistema de identificación de los planos de una imagen con los planos representados en un mapa global. Para ello, diferencian entre dos tipos de restricciones que

condicionan el emparejamiento. La primera se corresponde con características propias del plano, tales como la superficie del plano, la elongación de la nube de puntos y su color dominante. La segunda incluye relaciones geométricas entre parejas de planos. Estas relaciones involucran los ángulos entre parejas, así como su distancia, los centroides y la distancia ortogonal de un plano y el centroide del otro.

Resumiendo, para realizar el emparejamiento de planos, la mayoría de los métodos anteriormente citados se basa en la suposición de tener una transformación inicial correcta. Esto implica demasiada confianza en los procesos previos de cálculo, los cuales no suelen dar lugar a buenas estimaciones en entornos estructurados poco texturados. Por otra parte, los métodos que utilizan RANSAC requieren de al menos tres planos cuyas normales permitan cubrir $\mathbb{R}^3$. En caso de combinar planos con puntos, este proceso puede ser computacionalmente muy exigente. En [12], se presenta el único método que utiliza relaciones entre planos mediante grafos y no utiliza odometría. Sin embargo, este sistema solo evalúa relaciones entre parejas en lugar de evaluar un mayor número de relaciones entre planos, tal como hace nuestro método.

II-B. Estimación del movimiento y posición de la cámara

En los sistemas de odometría y SLAM visual, la trayectoria de la cámara puede ser calculada mediante varias técnicas, entre las cuales destacamos tres: los métodos basados en puntos característicos, los basados en la técnica de alineación de imágenes mediante el método directo, y los que se basan en ICP.

Los odómetros basados en puntos clave [13][14] minimizan el error de reproyección de puntos característicos previamente identificados en las dos imágenes. Para poder estimar el desplazamiento de una forma precisa, este método requiere que la imagen tenga abundancia de puntos característicos y que éstos sean distintivos. Los entornos estructurados carecen típicamente de estos puntos o no son suficientemente descriptivos para emparejarlos entre imágenes.

Por otra parte, los métodos directos o de foto-consistencia [15][16] minimizan el error fotométrico mediante la evaluación del nivel de intensidad de cada píxel de cada imagen. Aunque no necesitan ni extraer puntos clave ni hacer emparejamiento, este método requiere que la iluminación sea constante y que el punto de vista no varíe demasiado. Adicionalmente, necesitan una gran capacidad de computación por lo que suelen ejecutarse en unidades de procesamiento gráfico (GPU).

Otro método de registro de imágenes que ha demostrado obtener buen rendimiento es ICP [17]. ICP minimiza el error geométrico de los puntos 3D más cercanos entre dos imágenes. Este método sufre de problemas de mínimos locales cuando el movimiento de la cámara es rápido o hay un movimiento brusco entre imágenes consecutivas. Además, este método se basa solamente en formas geométricas distintivas; es decir, cuando la cámara se mueve en entornos estructurados donde solamente hay una pared, ICP tiende a un mínimo local.

Existen otros enfoques que no han sido tan populares como los anteriores. Por ejemplo, los métodos de registro basados en planos [6] son una buena solución debido a su abundancia en entornos estructurados. Además, son capaces de realizar el registro con menos requerimientos computacionales debido al inferior número de planos con respecto al número de puntos. Otra ventaja de su representación es que son capaces de filtrar ruido introducido por el sensor. Sin embargo, si no existen al menos tres planos cuyas normales abarcan $\mathbb{R}^3$, el uso únicamente de planos provoca un problema de degeneración.

Otro enfoque lo constituye los métodos basados en contornos [18]. Éstos funcionan muy bien en multitud de entornos debido a la normalmente abundante presencia de contornos, su repetitividad a la hora de detectarlos, su tolerancia a cambios de iluminación.

Otros enfoques utilizan combinaciones de métodos para realizar el registro. Estas combinaciones consiguen suplir las carencias de la utilización de cada método independientemente. Varios ejemplos de estas combinaciones son: líneas con puntos [19][20], puntos y planos [4] [10], líneas, puntos y planos [5][9], método semi-directo con planos [8], puntos con segmentos de planos [21], contornos con imágenes de profundidad [22], contornos con imágenes de profundidad e ICP punto-plano [23].

Debido a la robustez que ofrecen los planos a lo largo de múltiples imágenes, éstos son utilizados en múltiples trabajos como objeto de referencia en el mapa a lo largo de la trayectoria de la cámara. En lugar de ser utilizados para obtener la transformación imagen a imagen como hemos visto anteriormente, éstos son utilizados en procesos de optimización global [7][24][25]. En la mayoría de estos enfoques, el registro de imagen a imagen se suele calcular mediante sistemas de registro computacionalmente ligeros, obteniendo un registro preciso y rápido, que luego se acaba mejorando utilizando optimizadores o introduciendo restricciones del entorno en el mapa.

Para concluir, conviene decir que entre los métodos previamente revisados hay algunos que se adaptan a entornos estructurados y con poca textura, entre los que destacamos los basados en contornos, y los basados en primitivas de alto nivel como planos y líneas. No obstante, para evitar problemas de degeneración es necesario su combinación con otro método. Nosotros hemos optado por los contornos ya que evitan el problema anteriormente comentado de los planos. Además, el uso de planos permite realizar una buena estimación inicial de la transformación entre imágenes, necesaria para alinear los contornos entre imágenes cuando existe un movimiento rápido de la cámara.

III. Vista general del sistema

El sistema propuesto se basa en un registro de imagen a imagen mediante contornos. Cada vez que la calidad del registro entre imágenes es inferior a un umbral se crea una imagen clave. Es entonces cuando se estima la transformación entre las imágenes clave mediante la combinación de contornos y planos. El sistema propuesto comprende dos partes, las cuales llamaremos seguimiento y refinamiento de la posición de la cámara, tal como se muestra en la figura 2.

Para realizar el seguimiento de la cámara hemos utilizado una versión ligeramente modificada de REVO (Robust Edge Visual Odometry)[22][23]. REVO es un algoritmo de odometría visual que utiliza contornos y mapas de profundidad.

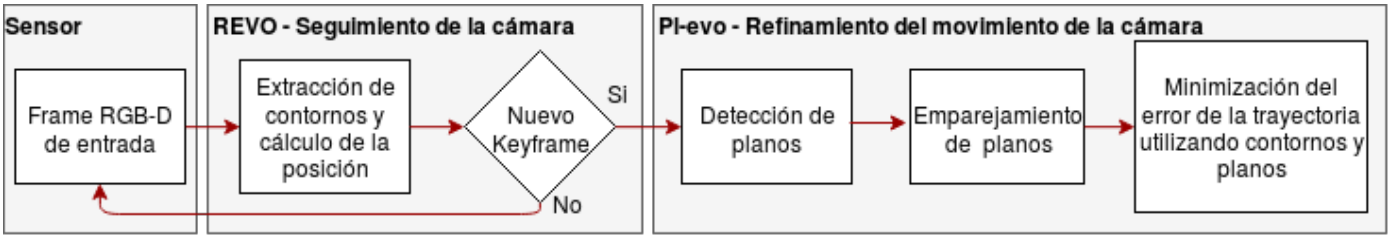


Figura 2: Vista general del sistema propuesto.

En este trabajo utilizamos una versión piramidal del detector de contornos de Canny [2], donde definimos tres niveles que resultan necesarios para realizar un registro adecuado de nivel de poco detalle a nivel de gran detalle. Además de la transformación obtenida entre posiciones de la cámara, REVO proporciona la detección de una nueva imagen clave en una secuencia de imágenes, la transformada distancia (DT) para la imagen actual, una nube de puntos, donde todos ellos son utilizados en la optimización de contornos.

Una vez se ha extraído una imagen clave, se procede a refinar la trayectoria entre imágenes clave consecutivas. Para ello se ha utilizado una combinación de contornos y planos. Para poder realizar esta combinación es necesario realizar previamente una extracción y una identificación de planos. Se ha utilizado los planos sólo en el proceso de refinamiento para evitar errores durante el proceso de seguimiento ya que implicaría reducir el número de imágenes procesadas por unidad de tiempo.

IV. Detección y emparejamiento de planos

Para realizar el cálculo de la transformación de una cámara en diferentes instantes de tiempo utilizando planos es necesario la detección y emparejamiento previo de estos planos entre diferentes imágenes. Para la detección se ha utilizado el algoritmo de agrupación jerárquica aglomerativa descrito en [26]. Este algoritmo detecta los planos de la escena utilizando la información de dos imágenes, una de color y una de profundidad, capturadas en un mismo instante de tiempo. Una vez detectados, para cada plano se extrae el color predominante, la ecuación normal del plano y la orientación. Estos parámetros son utilizados en la etapa de emparejamiento de planos, cuyo objetivo es identificar el mismo plano en instantes de tiempo diferentes.

Para identificar el mismo plano en dos fotogramas, Kf_r y Kf_c, se ha utilizado una versión modificada del método de emparejamiento de grafos descrito en [27], donde cada grafo alberga la información propia y la relación entre los planos detectados en un mismo fotograma. Dos matrices de afinidad son las encargadas de representar las similitudes entre los vértices (Kp) y las aristas (Kq) de los grafos, donde los vértices representan la información propia de los planos y las aristas representan las relaciones geométricas entres planos. La topología de cada grafo, la cual describe la conexión entre el punto inicial y el final de cada vértice, es representada mediante una matriz de incidencia G, donde G_1 describe el primer fotograma y G_2 el segundo. Otra matriz utilizada en este proceso es Ct, la cual representa los posibles candidatos de emparejamiento entre los dos grafos. Estas matrices son detalladas a continuación.

Kp alberga las similitudes de color entre cada uno de los planos de Kf_r y de Kf_c. Estas similitudes se calculan mediante la distancia de Bhattacharya entre histogramas de color. La distribución de color para cada plano se representa de forma submuestreada mediante la concatenación de histogramas de 16 niveles de intensidad para cada canal de color del espacio RGB.

Kq representa la similitud entre las relaciones geométricas de un plano con el resto de planos del mismo grafo Kf_r, y la misma característica del otro fotograma Kf_c. La relación geométrica entre planos se expresa como (1) la diferencia entre las orientaciones de los vectores normales respectivos entre planos del mismo fotograma y (2) la distancia entre ellos, siendo 0 si no son paralelos. Las similitudes entre los dos grafos se obtienen mediante la diferencia para cada una de esas distancias entre los dos grafos. Una vez obtenidos la similitud entre los grafos para cada distancia se obtiene la similitud final mediante una suma ponderada.

Ct reúne los posibles emparejamientos de planos entre grafos. Cada posición de la matriz Ct puede contener $\{0, 1\}$, donde $Ct_{i,j} = 1$ si es un emparejamiento candidato. Un plano de un fotograma es un candidato en el otro sólo si la orientación del plano, vertical, horizontal u oblicua, es la misma entre fotogramas.

La información de la correspondencia de planos entre fotogramas, así como las correspondientes ecuaciones del plano, se utilizan también en el proceso de estimación de la posición de la cámara.

V. Estimación de la posición de la cámara

En el odómetro visual propuesto, la entrada al sistema es una cámara RGB-D que suministra una imagen F_k para un instante de tiempo k. Cada imagen comprende una imagen de color I_c y una imagen de profundidad I_d donde cada píxel de ambas imágenes pertenece al mismo punto de la escena. El movimiento de la cámara entre dos instantes de tiempo se puede expresar mediante la transformación de un cuerpo rígido $T_{k,k-1} \in \mathbb{R}^{4\times4}$:

$$T_{k,k-1} = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix}, \quad (1)$$

donde $R_{k,k-1} \in SO(3)$ es la matriz de rotación, y $t_{k,k-1} \in \mathbb{R}^{3\times1}$ es el vector de traslación. El conjunto $T_{1:n} = T_{1_0}..., T_{n,n-1}$ expresa de forma compacta los desplazamientos

de la cámara entre las imágenes de 0 a n, mientras que $C_{0:n} = \{C_0, ..., C_n\}$ representa las posiciones de la cámara respecto a la posición inicial.

REVO calcula la transformación $T_{k,k-1}$ entre imágenes. Cuando una nueva imagen clave es detectada se calcula la transformación entre la imagen clave actual y la anterior T_r, T_c mediante la acumulación de las transformaciones calculadas para cada par de imágenes consecutivas. Esta transformación es utilizada como transformación inicial en el proceso de optimización de la trayectoria.

En este proceso de optimización se soluciona mediante un problema de mínimos cuadrados, donde combinamos información geométrica y fotométrica, planos y contornos respectivamente, mientras minimizamos el error de transformación entre imágenes clave, tal como se detalla en la ecuación 2:

$$E_{total} = E_{edge} + w_{pl} E_{planar}, \tag{2}$$

donde w_{pl} expresa el nivel de importancia relativa de los dos términos de error en la optimización, dando más importancia a los planos sobre los contornos. E_{edge} y E_{planar} se detallan en las siguientes secciones.

Para solventar el problema de mínimos cuadrados utilizamos el software Ceres [28], donde los parámetros a optimizar para ambos términos de la equación 2 son la rotación y traslación entre imágenes clave. En el caso de la rotación hemos utilizado la formulación de Rodrigues. Esta formulación nos aporta una representación eficiente de una rotación, planteándolo como la rotación de un ángulo θ alrededor de un vector unitario $v = [x, y, z]$. A diferencia de las representaciones basadas en los ángulos de Euler, esta formulación no tiene singularidades. Desafortunadamente presenta una de las desventajas de esta representación es que tiene una discontinuidad en π radianes. Este problema no aparece en nuestro sistema, ya que la rotación nunca llega a este valor debido al solapamiento entre imágenes. Nuestra formulación utiliza un vector unitario (x, y, z) y un ángulo θ, los cuales combinamos en un único vector $c_{rod} = [r_x, r_y, r_z] = [x/\theta, y/\theta, z/\theta]$, de forma que $\theta = 1/\sqrt{r_x^2 + r_y^2 + r_z^2}$. De esta forma, reducimos el número de parámetros a optimizar.

V-A. Minimización del error debido a los contornos

En el proceso de optimización, el término de error correspondiente a los contornos es calculado de una forma muy similar a la utilizada en REVO, con dos diferencias: por un lado, no utilizamos una estrategia piramidal ya que no necesitamos operar a diferentes niveles de detalle para ajustar la transformación inicial, porque ya la obtenemos mediante un emparejamiento de planos; y, por otro lado, ponderamos el error asociado a los píxeles de contorno en función de su distancia a la cámara, para contrarrestar la incertidumbre asociada a los puntos más alejados. Más concretamente, definimos un peso $w_d = p_z^{-2}$, donde p_z es la componente z del punto correspondiente de la escena.

Para calcular el error de los contornos r_e se calcula la distancia euclídea de un píxel de contorno de una imagen al píxel de contorno más cercano de la otra imagen según la ecuación 3. El contorno más cercano es obtenido mediante la evaluación de la transformada distancia DT para el fotográma clave de referencia.

$$E_{edge} = \sum_{p_e \in \Omega E_c} \delta_H(r_e) \cdot w_d \cdot r_e^2 \tag{3}$$

$$r_e = DT_r(\tau(T_{rc}, p_e, Z_c(p_e))) \tag{4}$$

$$\delta_H(r_e) = \begin{cases} 1 & \text{si } r_e \geq \Theta_H \\ 0 & \text{si } r_e < \Theta_H \end{cases} \tag{5}$$

En las ecuación es 3-5, r es la imagen de referencia y c es la imagen actual, Ω_{Ec} es el conjunto de píxeles con una profundidad valida en la imagen actual, τ calcula en que píxel de la imagen DT_r de la imagen clave de referencia es proyectado un píxel de contorno de la imagen clave actual p_e, donde esta proyección se calcula utilizando la transformación entre imágenes T_{rc} y la profundidad del píxel de contorno $Z_c(p_e)$, $\delta_H(r_e)$ es la función de perdida de Huber, la cual reduce la influencia de los errores grandes.

V-B. Minimización del error asociados a los planos

La idea general es minimizar la distancia entre planos de una imagen y sus correspondencias en la siguiente imagen. Muchos trabajos del estado del arte actual se basan en la diferencia a nivel de parámetros entre planos correspondientes. Desafortunadamente, este método no tolera bien el efecto producido por los puntos atípicos, ni tampoco el ruido de los puntos lejanos. Por esta razón, y como algunos trabajos ya realizan [5], hemos optado por un enfoque, el cual reproyecta los puntos de un plano de una imagen clave sobre el plano correspondiente de la otra imagen clave y acumula el error cometido. En nuestro método, en lugar de evaluar todos los puntos del plano, se ha optado por utilizar los puntos frontera del segmento correspondiente, ya que estos muestran una representación simplificada de su geometría y consiguen minimizar la cantidad de puntos a evaluar.

Denotamos la ecuación de un plano j en una imagen clave como $\pi_j = (a_j, b_j, c_j, d_j)$ donde (a_j, b_j, c_j) es la normal del plano n_j y d_j es la distancia del plano al centro óptico de la cámara.

El error asociado a los puntos frontera del plano y su correspondencia se denota como r_{pl} y se obtiene de la distancia perpendicular entre los puntos frontera de un plano de la imagen clave de referencia proyectado en la imagen clave actual. Esto se corresponde con la ecuación 7.

$$E_{planar} = \sum_{v_i \in Pl_{(j,r)}} \sum_{\pi_k \in Pl_c} S \cdot \delta_C(r_{pl}) \cdot w_d \cdot r_{pl}^2 \tag{6}$$

$$r_{pl} = n_{(j,c)}^T \cdot R_{rc} \times v_{(i,r)} + n_{(j,c)}^T \cdot T_{rc} - d_{(j,c)} \tag{7}$$

$$\delta_C(r_{pl}) = log(1 + r_{pl}) \tag{8}$$

En las ecuaciones 6-8 v_i es el conjunto de puntos frontera de un plano en la imagen de referencia $Pl_{j,r}$. π_k representa la ecuación del plano correspondiente a un plano de la imagen actual Pl_c, la correspondencia entre planos esta representada por $S \epsilon \{0,1\}$, donde $S_{v_i,\pi_k} = 1$ si un punto frontera de un plano de la imagen de referencia tiene correspondencia con el plano de la imagen actual, $\delta_C(r_{pl})$ se corresponde con la función de Cauchy (ecuación 8) y se encarga de filtrar los errores de los valores atípicos, w_d tiene la misma funcionalidad y se calcula de la misma forma que en la sección V-A, R_{rc} y T_{rc} se corresponde respectivamente con la rotación y traslación entre imágenes clave.

VI. Resultados y experimentos

En esta sección se evalúa nuestro odómetro visual comparándolo con otros odómetros visuales y también con métodos de SLAM. Se ha utilizado dos conocidos datasets públicos que incluyen las trayectorias estimadas como referencia para su evaluación: el banco de pruebas de TUM RGB-D [29] y el ICL-NUIM [30], que a diferencia del anterior, utiliza escenas sintéticas libres de ruido. Se ha utilizado la herramienta de evaluación de TUM RGB-D [29] para comparar las posiciones de las cámaras de nuestro odómetro con su correspondiente valor de referencia. La métrica usada para compararnos con otros métodos es el error cuadrático medio RMSE (Root-Mean-Square Error).

Evaluando los resultados obtenidos, cabe destacar que se está evaluando un odómetro visual con otros odómetros y métodos de SLAM. A diferencia de los métodos de SLAM, los odómetros no son capaces de relocalizarse una vez se ha perdido el registro ni de mejorar el error acumulado a lo largo de la trayectoria de la cámara mediante la restricción de cierre de bucle. Por ejemplo, en el caso de los datasets *fr3_str_notex_rear*, *fr3_str_notex_far* y *ICL/office1* los métodos de ORB-SLAM[13] y LSD-SLAM [31] se pierden debido a un error de registro. En el caso de EDGE-SLAM aun habiéndose perdido, es capaz de recuperar la posición gracias a la relocalización. Cabe destacar que el método propuesto es capaz de realizar toda la trayectoria sin que falle el registro, lo cual es una gran ventaja respecto a los otros métodos.

En la tabla I se muestran los valores de RMSE del método propuesto Pl-EVO, LSD-SLAM[31], ORB-SLAM [13], PL-SLAM [19], REVO [22], Edge-SLAM y Edge-VO [18]. En REVO se pueden diferenciar dos columnas, la primera de ella muestra los resultados obtenidos utilizando imágenes clave con la configuración disponible online, mientras que los resultados de la columna $E+D+Opt$ están extraídos directamente de su artículo. A diferencia del algoritmo que nosotros utilizamos, el propuesto en [22] realiza una optimización de la posición de la imagen clave actual con N previos. Utilizando este concepto, consiguen reducir la deriva del cálculo de la posición de la cámara y cerrar pequeños bucles.

Analizando la tabla I, se puede observar que el método propuesto obtiene mejores resultados en entornos estructurados y con poca textura que los mejores métodos de SLAM del estado del arte actual sin realizar relocalización, ni cierre de bucles y solo realizando optimización entre dos imágenes consecutivas. Además, se puede observar como se obtienen resultados muy similares al resto en entornos no estructurados.

Además del cálculo de la trayectoria de la cámara, con el método propuesto se generan mapas densos del entorno mediante una versión ligeramente modificada de la existente en REVO. Esta representación sólo utiliza la inserción de imágenes clave en el mapa para reducir la cantidad de información. En la figura 3, se puede observar las diferencias entre uno de los mapas obtenidos utilizando la odometría de la versión online de REVO y nuestro método. Tal como se muestra en la figura 3, el mapa se aleja de la realidad debido a un error de registro en varias imágenes clave consecutivas.

VII. Conclusiones

En este trabajo se ha descrito un método de odometría visual. Con este método se consigue solventar el fallo de registro entre imágenes debido a la ausencia de textura en los métodos de odometría y SLAM visual actuales, consiguiendo mejores resultados en entornos estructurados y resultados muy similares en los demás entornos. Estas mejoras se han conseguido mediante la optimización entre imágenes clave utilizando contornos y planos. El número de contornos detectados en entornos con poca textura es superior a los puntos de interés. Por ello, Pl-EVO consigue una mayor precisión en la trayectoria de la cámara.

REVO falla en el registro cuando hay mucho desplazamiento entre imágenes. Introduciendo planos en la optimización conseguimos por una parte filtrar puntos erróneos y por otra parte estimar una mejor transformación inicial.

En este artículo también se ha presentado un método de emparejamiento de planos, el cual a diferencia de los métodos actuales, no se basa en el cálculo previo de transformaciones entre imágenes, ni utiliza RANSAC para calcular la transformación entre imágenes. El método propuesto sólo utiliza restricciones propias de los planos y, a diferencia de los otros métodos, establece relaciones entre ellos mediante grafos. Con esto se consigue realizar el emparejamiento cuando la transformación entre imágenes es errónea, o por otra parte cuando hay muy pocos planos para estimar un modelo con RANSAC.

El siguiente paso es optimizar la trayectoria de la cámara utilizando la información de múltiples imágenes clave, y la representación de los planos con su posición global. Además se quiere identificar planos previamente vistos mediante una versión modificada del método de emparejamiento de planos presentado en este artículo.

VIII. Agradecimientos

Este trabajo ha sido parcialmente financiado por el proyecto ROBINS (EU-H2020, el proyecto español MINECO DPI2014-57746-C3-2-R GA 779776), PGC2018-095709-B-C21 (MCIU/AEI/FEDER, UE), PROCOE/4/2017 (Govern Balear, 50 % P.O. FEDER 2014-2020 Illes Balears) y la beca BES-2015-071804.

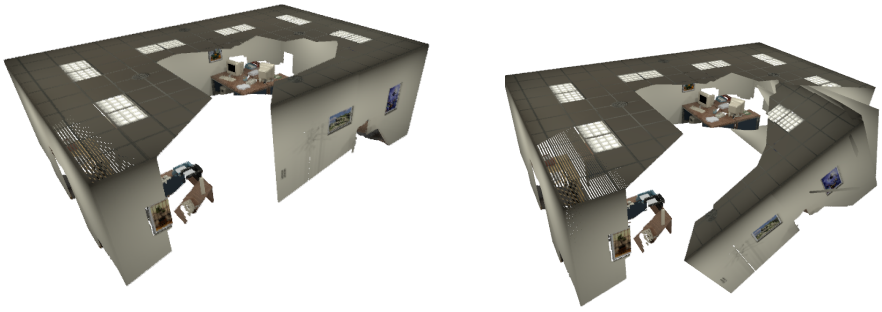
Referencias

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. [Online]. Available: https://doi.org/10.1023/B:VISI.0000029664.99615.94

Tabla I: Error absoluto de la trayectoria calculada (RMSE, cm)

Secuencia	Pl-EVO	REVO	Edge SLAM †	ORB-SLAM†	LSD-SLAM†	Edge VO†	PL-SLAM†	REVO E+D+Opt‡
fr1/xyz*	3.56	4.86	1.31	**0.9**	9.0	16.51	1.21	1.55
fr2/xyz*	1.50	1.90	0.49	**0.3**	2.15	21.41	0.43	-
fr3/str_notex_far	**1.60**	2.38	6.71	×	×	41.76	×	2.17
ICL/office0*	**2.78**	6.70	3.21	5.67	×	×	-	-
ICL/office1	1.03	×	19.5	×	×	×	×	**0.98**
ICL/office3*	**0.89**	2.96	4.58	16.18	×	×	×	-

– Significa que la información no está disponible.
× Muestra los casos en los que no se ha podido inicializar o se ha perdido en múltiples ejecuciones.
† Los resultados de Edge SLAM, ORB-SLAM LSD-SLAM y Edge VO se han extraído de [18].
‡ El resultado de REVO E+D+Opt se ha extraído de [23]
∗ Indica que en el dataset evaluado se podrían detectar cierre de bucles.



(a) Pl-EVO (b) REVO

Figura 3: Mapa generado por Pl-EVO y por REVO.

[2] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov 1986.

[3] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[4] Y. Taguchi, Y. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3D sensors," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 5182–5189.

[5] M. Dou, L. Guan, J.-M. Frahm, and H. Fuchs, "Exploring high-level plane primitives for indoor 3D reconstruction with a hand-held RGB-D camera," in *Computer Vision - ACCV 2012 Workshops*, J.-I. Park and J. Kim, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 94–108.

[6] K. Pathak, A. Birk, N. Vaskevicius, and J. Poppinga, "Fast registration based on noisy planes with unknown correspondences for 3D mapping," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 424–441, June 2010.

[7] R. F. Salas-Moreno, B. Glocken, P. H. J. Kelly, and A. J. Davison, "Dense planar SLAM," in *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Sept 2014, pp. 157–164.

[8] M. Hsiao, E. Westman, G. Zhang, and M. Kaess, "Keyframe-based dense planar SLAM," in *IEEE International Conference on Robotics and Automation, ICRA, Singapore*, 2017.

[9] P. F. Proença and Y. Gao, "Probabilistic combination of noisy points and planes for RGB-D odometry," *CoRR*, vol. abs/1705.06516, 2017. [Online]. Available: http://arxiv.org/abs/1705.06516

[10] E. Ataer-Cansizoglu, Y. Taguchi, S. Ramalingam, and T. Garaas, "Tracking an RGB-D camera using points and planes," in *2013 IEEE International Conference on Computer Vision Workshops*, Dec 2013, pp. 51–58.

[11] Y. Shi, K. Xu, M. Niessner, S. Rusinkiewicz, and T. Funkhouser, "Planematch: Patch coplanarity prediction for robust RGB-D reconstruction," *arXiv preprint arXiv:1803.08407*, 2018.

[12] E. Fernández-Moral, P. Rives, V. Arévalo, and J. González-Jiménez, "Scene structure registration for localization and mapping," *Robotics and Autonomous Systems*, vol. 75, pp. 649 – 660, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889015001979

[13] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-SLAM: A versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct 2015.

[14] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an RGB-D camera," in *In Proc. of the Intl. Sym. of Robot. Research*, 2011.

[15] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[16] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. Citeseer, 2013, pp. 2100–2106.

[17] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, Oct 2011, pp. 127–136.

[18] S. Maity, A. Saha, and B. Bhowmick, "Edge SLAM: Edge points based monocular visual SLAM," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, Oct 2017, pp. 2408–2417.

[19] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: Real-time monocular visual SLAM with points and lines," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 4503–4508.

[20] X. Zuo, X. Xie, Y. Liu, and G. Huang, "Robust visual SLAM with point and line features," *arXiv preprint arXiv:1711.08654*, 2017.

[21] R. Li, Q. Liu, J. Gui, D. Gu, and H. Hu, "A novel RGB-D SLAM algorithm based on points and plane-patches," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug 2016, pp. 1348–1353.

[22] F. Schenk and F. Fraundorfer, "Combining edge images and depth maps for robust visual odometry," in *Proc. 28th British Machine Vision Conference*, 2017, pp. 1–12.

[23] F. Schenk and F. Fraundorfer, "Robust edge-based visual odometry using machine-learned edges," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 1297–1304.

[24] L. Ma, C. Kerl, J. Stückler, and D. Cremers, "CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1285–1291.

[25] M. Kaess, "Simultaneous localization and mapping with infinite planes," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 4605–4611.

[26] C. Feng, Y. Taguchi, and V. R. Kamat, "Fast plane extraction in organized point clouds using agglomerative hierarchical clustering," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 6218–6225.

[27] F. Zhou and F. De la Torre, "Factorized graph matching," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 9, pp. 1774–1789, 2016.

[28] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[29] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 573–580.

[30] A. Handa, T. Whelan, J. McDonald, and A. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Hong Kong, China, May 2014.

[31] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 834–849.