

Hacia un sistema SLAM Visual e Inercial con Estimación Consistente de Incertidumbre en Tiempo Acotado

Gastón Castro
Universidad de San Andrés
CONICET-UDESA
Buenos Aires, Argentina
gcastro@udesa.edu.ar

Pablo Alcantarilla
SLAMcore
Londres, Reino Unido
pablo@slamcore.com

Taihú Pire
Laboratorio de Robótica
CIFASIS (CONICET-UNR)
Rosario, Argentina
pire@cifasis-conicet.gov.ar

Abstract—

Este trabajo aborda el problema de SLAM Visual e Inercial con el objetivo de presentar una solución consistente en términos de estimación de la incertidumbre. Se propone un método capaz de calcular incertidumbres globales aproximadamente consistentes sin aumentar la complejidad con el tamaño total del área explorada. El método permite la selección arbitraria de áreas locales para optimización, introduciendo una metodología para construir un *prior* virtual en tiempo acotado. Se utiliza una topología de factores aproximando el *prior* utilizando árboles de Chow-Liu, obteniendo un problema de optimización local que permite calcular estimaciones de incertidumbre consistentes. Esto supera varias deficiencias de enfoques anteriores que se basan en condicionamiento (fijación de variables) y/o marginalización con ventana deslizante. El trabajo se basa en el sistema de SLAM Visual descrito en [1] con las siguientes contribuciones: i) tratamiento de mediciones inerciales, ii) experimentos basados en un nuevo framework de simulación, iii) una implementación eficiente en C++.

Index Terms—SLAM, Estimación de estado, Optimización, Incertidumbre, Consistencia, Marginalización

I. INTRODUCCIÓN

En un mundo donde a los robots se les asignan periódicamente tareas cada vez más diversas, resulta esencial proporcionar capacidades de localización y mapeo eficientes, sólidas y precisas. Este problema se denomina localización y mapeo simultáneos (SLAM), y es un requisito fundamental para lograr una navegación autónoma, y representa uno de los principales temas de investigación en la comunidad de robótica [2].

Un método de SLAM capaz de trabajar en el mundo real requiere manejar medidas con diversos grados de incertidumbre y error debido a entornos impredecibles, sensores contaminados por ruido, representaciones discretas entre otras. Por tanto, es deseable que los sistemas SLAM proporcionen una medida de incertidumbre contenida en cada variable estimada. El consenso es tratar estas propiedades físicas (poses de los robots y puntos de referencia) como variables aleatorias que siguen una determinada distribución probabilística [3], [4], [5], con formulaciones muy novedosas entre las que destaca [6].

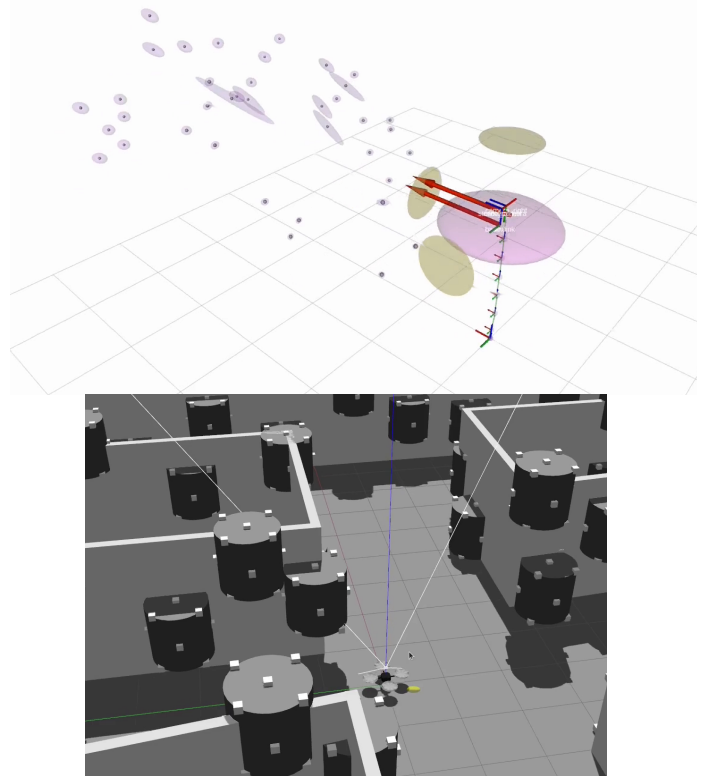


Fig. 1: Método de estimación funcionando en un escenario simulado. El método estima las variables de estado de un dron simulado, los *landmarks* del mapa y su incertidumbre representada como elipsoides en color magenta. La imagen inferior muestra el entorno de simulación en Gazebo con *landmarks* simulados.

Como se define en [7], un estimador de estado se considera *consistente* si los errores de estimación son de media cero y la covarianza estimada es mayor o igual, en el sentido matricial positivo-definido, a la covarianza calculada mediante un enfoque óptimo dado, como la resultante de optimizar el del grafo completo (*batch-optimization*). La *batch-optimization* como línea base de consistencia también tiene una mayor significancia teórica, ya que la matriz Hessiana resultante del sistema se aproxima a la matriz de información de Fisher, que

a su vez es el límite inferior de Cramér-Rao (CRLB) [5].

En este trabajo se propone un método de SLAM Visual e Inercial capaz de lograr covarianzas consistentes. El método se basa en la construcción de *priors* virtuales, que son representaciones simplificadas del mapa global. Estos *priors* se utilizan para contextualizar las optimizaciones locales, lo que resulta en estimaciones de incertidumbre más consistentes. En la Figura 1 se puede observar un escenario donde el método estima las variables de estado de un dron simulado, los landmarks del mapa y las incertidumbre involucradas en las estimaciones (representadas como elipsoides de confianza).

II. TRABAJO RELACIONADO

Lograr un estimador de estado consistente que no aumente en complejidad con el tamaño del problema es un área de gran interés científica. Varios enfoques pueden contener la escala de complejidad del problema SLAM, pero caen en metodologías con inconvenientes que generan inestabilidades a futuro. Se pueden mencionar principalmente dos tipos de enfoques de complejidad constante o acotada, para mantener la operación en tiempo real: (a) métodos que realizan optimizaciones de área local mediante acondicionamiento, también conocido como fijación de variables; y (b) métodos que realizan optimizaciones de ventana deslizante marginalizando toda información previa.

Es posible realizar una optimización local sobre un área del grafo condicionando las variables seleccionadas a aquellas directamente relacionadas que componen la frontera del área. El principal supuesto es que las estimaciones de las variables fijas ya han convergido y son correctas. Varios trabajos de investigación en Visual e Inercial SLAM han reportado un gran éxito utilizando optimización y condicionamiento local [8], [9], [10], [11]. Un inconveniente predominante de este enfoque es que, como se supone que las variables fijas son completamente certeras, no es posible recuperar estimaciones de incertidumbre globalmente consistentes. Debe haber un factor (llamado factor *prior*) en el grafo local que responda a las distribuciones de probabilidad marginal y modele la correlación entre las variables seleccionadas, impuestas por las mediciones omitidas.

En el caso de una solución de ventana deslizante, es posible eliminar adecuadamente las variables que quedan fuera de una ventana de estimación de tamaño fijo; este proceso se conoce como “marginalización de variables”. La información recopilada se integra recursivamente y se transporta en un factor de distribución *prior* que correlaciona las variables activas dentro de la ventana de estimación [12], [13]. Qin et al. [11] traslada la ventana de estimación y continua usando el *prior* construido hasta ese momento, incluso después del cierre de un ciclo. En [14] se descarta el *prior* y construye uno nuevo usando condicionamiento como medio para cortar la historia anterior del grafo pero, aunque esto es posible, la consistencia de la incertidumbre global no se mantiene.

Hay tres consideraciones importantes sobre este enfoque: (i) conlleva densificar matrices originalmente dispersas; (ii)

surgen problemas de linealización a medida que los modelos jacobianos se integran y nunca se vuelven a re-calcular; y (iii) el factor *prior* se vuelve inexacto, o incluso inválido, cuando se cierra un ciclo en la trayectoria, ya que significa nueva información relacionada con variables ya marginalizadas. Esto implica que los enfoques de marginalización de ventana deslizante solo son capaces de proporcionar estimaciones consistentes de incertidumbre que pueden deteriorarse progresivamente.

El presente trabajo propone un nuevo enfoque de optimización local para SLAM que aborda todos los inconvenientes antes mencionados y proporciona una solución que calcula las incertidumbres globales manteniendo las ventajas computacionales de las optimizaciones locales. El método permite la selección de áreas locales arbitrarias para la actualización preservando al mismo tiempo una consistencia aproximada tanto en los valores estimados como en las incertidumbres. La solución define una topología de factores virtuales aproximando la covarianza marginal del área local utilizando el algoritmo de Chow-Liu [15]. Se destaca la inclusión del tratamiento de mediciones inerciales siendo un incremental sobre un trabajo previamente publicado [1].

III. DEFINICIÓN DEL PROBLEMA

El problema de estimación SLAM aumenta su número de variables a medida que el robot navega por el entorno y explora. En un determinado momento de la trayectoria, las variables del problema se definen como $\mathbf{x} = \{\mathbf{x}_r, \mathbf{x}_l\}$ donde $\mathbf{x}_r = \{\mathbf{x}_{r_1}, \dots, \mathbf{x}_{r_n}\}$ representa n estados del robot y $\mathbf{x}_l = \{\mathbf{x}_{l_1}, \dots, \mathbf{x}_{l_m}\}$ representa m marcas de referencia (*landmarks*) del entorno. El conjunto de mediciones se indicará como $\mathbf{z} = \{\mathbf{z}_{ij}\} \cup \{\mathbf{u}_{ij}\}$ donde \mathbf{z}_{ij} es una observación del *landmark* j desde la perspectiva del robot en el estado i y \mathbf{u}_{ij} es un incremento relativo medido entre los estados del robot i y j . Es posible obtener una solución de probabilidad máxima a posteriori (*maximum a posteriori probability*, MAP) para las variables de estado maximizando una distribución de probabilidad objetivo compuesta por las probabilidades de las mediciones y un término previo (*prior*):

$$\begin{aligned} \hat{\mathbf{x}}^{\text{MAP}} &= \underset{\mathbf{x}}{\operatorname{argmax}} p(\mathbf{x}|\mathbf{z}) = \underset{\mathbf{x}}{\operatorname{argmax}} \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z})} \\ &= \underset{\mathbf{x}}{\operatorname{argmax}} p(\mathbf{z}|\mathbf{x})p(\mathbf{x}), \end{aligned} \quad (1)$$

donde el factor de normalización $p(\mathbf{z})$ se puede eliminar ya que no afecta la maximización, $p(\mathbf{z}|\mathbf{x})$ son las funciones de verosimilitud de las mediciones y $p(\mathbf{x})$ el término *prior*. El modelo basado en grafos de factores (*Factor Graphs*) [4] puede ser utilizado para representar una factorización eficientemente de la distribución objetivo y resolver la maximización. Siendo $\mathcal{G} = (\mathcal{F}, \mathcal{X}, \mathcal{E})$ un grafo de factores bipartito, con dos tipos de nodos: factores $f_k \in \mathcal{F}$ y variables $\mathbf{x}_i \in \mathcal{X}$; las aristas $e_{ki} \in \mathcal{E}$ relacionan factores y variables y el conjunto de variables adyacentes a un factor f_k se notará como \mathbf{x}_k . Luego,

el grafo de factores modela una factorización de una función de costo global como:

$$F(\mathbf{x}) = \prod_{k \in \mathcal{G}} f_k(\mathbf{x}_k), \quad (2)$$

con cada f_k siendo una función proporcional (denotada con \propto),

$$f_k(\mathbf{x}_k) \propto \left\{ \exp\left(-\frac{1}{2}\|\mathbf{z}_k \boxminus m_k(\mathbf{x}_k)\|_{\Sigma_k}^2\right) \right\}, \quad (3)$$

donde \boxminus el meta-operador que calcula el término de error entre las mediciones \mathbf{z}_k y los valores esperados $m_k(\mathbf{x}_k)$, con las covarianzas Σ_k siendo las incertidumbres de los errores y m_k una función de modelo relacionada al factor f_k . Estos parámetros son parte de cada factor y están definidos por los siguientes tres modelos:

El modelo de observación describe las mediciones esperadas del sensor, al observar el *landmark* j desde la pose del robot i , se define $\mathbf{z}_{ij} = h_{ij}(\mathbf{x}_{r_i}, \mathbf{x}_{l_j}) + \eta_{ij}$ donde $\eta_{ij} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{ij})$ y \mathbf{R}_{ij} es la matriz de covarianza del modelo de observación.

El modelo de movimiento describe el incremental de movimiento esperado entre dos estados de la trayectoria, se define $\mathbf{u}_{ij} = g_{ij}(\mathbf{x}_{r_i}, \mathbf{x}_{r_j}) + \omega_{ij}$ el incremento entre el estado i y el estado j donde ω_{ij} es un vector de ruido aditivo que sigue una distribución normal $\omega_{ij} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{ij})$ y \mathbf{Q}_{ij} representa la incertidumbre.

El término prior refleja un conocimiento previo que se tiene sobre un subconjunto de las variables de estado, ya sea a partir de información de inicialización o de un proceso de marginalización. Se asume que este término *prior* tiene una distribución normal e impone una expectativa sobre las variables asociadas, es decir, $\mathbf{x}_p \sim \mathcal{N}(\mathbf{x}_\Pi, \Sigma_\Pi)$ donde $\mathbf{x}_p \in \mathbf{x}$.

Aplicando logaritmo negativo, la estimación máxima a posteriori en la Eq. (1) se reduce a minimizar la suma de todos los errores de predicción asociados a los factores del grafo:

$$\hat{\mathbf{x}}^{\text{MAP}} = \underset{\mathbf{x}}{\operatorname{argmin}} F(\mathbf{x}) = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{k \in \mathcal{G}} \|\mathbf{z}_k \boxminus m_k(\mathbf{x}_k)\|_{\Sigma_k}^2, \quad (4)$$

donde relajamos la notación, siendo $\mathbf{z}_k \in \{\mathbf{z}\} \cup \{\mathbf{x}_\Pi\}$ y m_k representa cualquier modelo de función explicado previamente: observación, movimiento o *prior*. Los errores de predicción asociados a cada factor también se los denota cómo residuos $\mathbf{r}_k = \mathbf{z}_k \boxminus m_k(\mathbf{x}_k)$, y la minimización de la Eq. (4) se la conoce como un problema de optimización de mínimos cuadrados no lineal para el cual existen varias implementaciones altamente eficientes para resolverlo [16], [17], [18]. Estos métodos se basan en la linealización iterativa de los términos de error no lineales aplicando expansiones de Taylor de primer orden, obteniendo un sistema lineal que resuelve una actualización incremental $\Delta\mathbf{x}$ en el vector de estado \mathbf{x} que se aproxima la solución máxima a posteriori:

$$\begin{aligned} \mathbf{H}\Delta\mathbf{x}^* &= -\mathbf{b} \\ \hat{\mathbf{x}}^{\text{MAP}} &\approx \mathbf{x} \boxplus \Delta\mathbf{x}^*. \end{aligned} \quad (5)$$

El estado \mathbf{x} (o parte de él) generalmente no pertenece a un espacio euclídeo (por ejemplo, la orientación en poses 2D/3D),

y es un enfoque común realizar la optimización del incremento $\Delta\mathbf{x}$ en un espacio tangente que sí lo sea. El meta-operador \boxplus modela la composición entre ambos espacios [19]. Luego, es posible calcular la incertidumbre asociada a la estimación invirtiendo la matriz \mathbf{H} , conocida como matriz Hessiana:

$$\hat{\Sigma}^{\text{MAP}} \approx \mathbf{H}^{-1} \quad \text{with} \quad \mathbf{H} = \mathbf{J}^T \mathbf{\Omega} \mathbf{J}, \quad (6)$$

donde \mathbf{J} es la matriz jacobiana compuesta por las derivadas de los términos de error, y $\mathbf{\Omega}$ es una matriz diagonal de bloques compuesta por $\mathbf{\Omega}_k = \Sigma_k^{-1}$. La complejidad computacional de esta solución canónica aumenta ilimitadamente con el tamaño, cada vez mayor, del problema y no es un enfoque eficiente para sistemas de ejecución en tiempo real.

A. Modelo visual e inercial

En esta sección se definen las variables involucradas, así como los modelos de observación y movimiento para el problema de SLAM Visual e Inercial. Cada estado i del robot se divide entonces como $\mathbf{x}_{r_i} = \{\mathbf{T}_i, \mathbf{v}_i, \mathbf{b}_i^g, \mathbf{b}_i^a\}$ siendo $\mathbf{T}_i = [\mathbf{R}_i, \mathbf{p}_i] \in \mathbb{SE}(3)$ la pose del robot y $\mathbf{v}_i \in \mathbb{R}^3$ su velocidad lineal instantánea en el sistema de coordenadas del "mundo", y $\mathbf{b}_i^g \in \mathbb{R}^3$, $\mathbf{b}_i^a \in \mathbb{R}^3$ los bias del giroscopio y acelerómetro respectivamente los cuales se asumen que siguen un modelo de movimiento Browniano [20].

Considerando que los elementos visuales detectados en las imágenes corresponden a elementos en el espacio tridimensional, cada *landmark* $\mathbf{x}_{l_j} \in \mathbb{R}^3$ y los residuos asociados a los factores visuales están dados por

$$\mathbf{r}_{ij} = \mathbf{z}_{ij} - \operatorname{Prj}(\mathbf{T}_{CB} \mathbf{T}_i \mathbf{x}_{l_j}), \quad (7)$$

donde $\operatorname{Prj} : \mathbb{R}^3 \rightarrow \mathbb{R}^n$ es la función de proyección correspondiente al modelo de la cámara, $\mathbf{z}_{i,j} \in \mathbb{R}^n$ y $\mathbf{T}_{CB} \in \mathbb{SE}(3)$ corresponde a la rígida entre el centro de coordenadas del cuerpo del robot (usualmente centrado en la IMU) y el sistema de coordenadas de la cámara.

En cuanto a los factores de movimiento, las mediciones inerciales de aceleración lineal y velocidad de rotación obtenidas de la IMU entre dos instantes de la trayectoria son preintegradas utilizando el framework de preintegración [20]. Esto permite obtener incrementales de pose y velocidad en el espacio de las variables del robot, es decir, $\mathbf{u}_{i,i+1} = [\Delta\mathbf{R}_{i,i+1}, \Delta\mathbf{v}_{i,i+1}, \Delta\mathbf{p}_{i,i+1}]$, integrando las mediciones del giroscopio y del acelerómetro considerando sus bias. El framework además resuelve una matriz de covarianza $\mathbf{Q}_{i,i+1}$ que modela la incertidumbre acumulada de las mediciones. En base a estos incrementales es posible definir los siguientes residuos,

$$\begin{aligned} \mathbf{r}_{i,i+1} &= [\mathbf{r}_{\Delta\mathbf{R}_{i,i+1}}, \mathbf{r}_{\Delta\mathbf{v}_{i,i+1}}, \mathbf{r}_{\Delta\mathbf{p}_{i,i+1}}, \mathbf{r}_{\mathbf{b}_{i,i+1}^g}, \mathbf{r}_{\mathbf{b}_{i,i+1}^a}] \\ \mathbf{r}_{\Delta\mathbf{R}_{i,i+1}} &= \operatorname{Log}(\Delta\mathbf{R}_{i,i+1}^T \mathbf{R}_i^T \mathbf{R}_{i+1}^T) \\ \mathbf{r}_{\Delta\mathbf{v}_{i,i+1}} &= \Delta\mathbf{v}_{i,i+1} - \mathbf{R}_i^T (\mathbf{v}_{i+1} - \mathbf{v}_i - \mathbf{g} \Delta t_{i,i+1}) \\ \mathbf{r}_{\Delta\mathbf{p}_{i,i+1}} &= \Delta\mathbf{p}_{i,i+1} - \mathbf{R}_i^T (\mathbf{p}_{i+1} - \mathbf{p}_i - \mathbf{v}_i \Delta t_{i,i+1} - \frac{1}{2} \mathbf{g} \Delta t^2) \\ \mathbf{r}_{\mathbf{b}_{i,i+1}^g} &= \mathbf{b}_{i+1}^g - \mathbf{b}_i^g \\ \mathbf{r}_{\mathbf{b}_{i,i+1}^a} &= \mathbf{b}_{i+1}^a - \mathbf{b}_i^a, \end{aligned} \quad (8)$$

donde \mathbf{g} es la dirección del vector gravedad, $\Delta t_{i,i+1}$ es el tiempo transcurrido entre las poses y $\text{Log} : \text{SO}(3) \rightarrow \mathbb{R}^3$ la función que relaciona elementos del grupo de Lie a un espacio vectorial.

IV. OPTIMIZACIÓN LOCAL CONSISTENTE

El objetivo del método propuesto es permitir la selección y optimización de áreas locales del grafo de factores manteniendo la consistencia probabilística del resultado agregando a la optimización una topología de factores *prior* que la contextualizan en términos probabilísticos de valores esperados e incertidumbre. Para esto, el método se centra en mantener una matriz de covarianza aproximada del sistema para la cual se relajan sus requisitos estructurales, dado que esta nueva estructura de almacenamiento de incertidumbre no será el resultado de un estimador adecuado como el que se obtiene de la inversión del Hessiano del sistema completo (Eq. 6). Es posible que la covarianza aproximada no mantenga la propiedad semidefinida positiva, pero mantendrá un mayor grado de dispersión (*sparsity*). Por todas estas razones, se la denomina **matriz de covarianza aproximada** y se anotará como $\tilde{\Sigma}$.

Algorithm 1 Optimización Local Consistente

Input: \mathcal{G} : complete factor graph,
 $\hat{X}, \tilde{\Sigma}$: mean and approximated covariance
Output: $\hat{X}, \tilde{\Sigma}$: updated mean and approx. covariance

```

1:  $\hat{X}^+, \mathcal{G}^+ \leftarrow \text{selectLocalArea}(\mathcal{G}, \hat{X}, \tilde{\Sigma})$ 
2:  $\mathcal{P} \leftarrow \text{constructVirtualPrior}(\mathcal{G}, \hat{X}^+, \tilde{\Sigma})$ 
3:  $\hat{X}^+, \mathbf{H}^+ \leftarrow \text{optimizeGraph}(\hat{X}^+, \mathcal{G}^+ \cup \mathcal{P})$ 

4:  $\hat{X}_i \leftarrow \hat{X}^+ \quad \forall i \in \text{variables}(\mathcal{G}^+)$ 
5:  $\tilde{\Sigma}_{ij} \leftarrow [\mathbf{H}^+]^{-1} \quad \forall i, j \in \text{variables}(\mathcal{G}^+)$ 

```

El Alg. 1 exhibe una descripción general del método propuesto que está diseñado de manera flexible para ser aplicado en una variedad de problemas SLAM. Cómo entrada requiere el grafo de factores completo \mathcal{G} , la última estimación de las variables \hat{X} y la covarianza aproximada $\tilde{\Sigma}$. El método se divide en tres pasos principales: En la línea (1) se selecciona un subgrafo de factores $\mathcal{G}^+ \subseteq \mathcal{G}$ y su conjunto de variables correspondientes \hat{X}^+ definiendo el área de optimización local; la línea (2) utiliza la covarianza aproximada $\tilde{\Sigma}$ para construir un conjunto de factores *prior* \mathcal{P} con una topología predefinida la cual representa la correlación existente entre las variables seleccionadas \hat{X}^+ ; y la línea (3) optimiza el subgrafo de variables y factores seleccionados con la topología *prior* \mathcal{P} agregada. Finalmente, solo los bloques correspondiente de la matriz de covarianza aproximada se actualizan con la matriz Hessiana resultante de la optimización local (líneas 4-5).

Cabe destacar que la complejidad del método de optimización local está determinada en última instancia por la conectividad del área local seleccionada y no por el tamaño del grafo completo. El sistema fue desarrollado en C++, utilizando ROS y el *framework* de optimización GTSAM [21]. En las

siguientes secciones, se abordará en mayor detalle cada paso del método.

A. Selección de Área local

El objetivo es definir un conjunto de variables X^+ y un subgrafo de factores \mathcal{G}^+ que determine el área de interés local. El conjunto completo de variables se dividirá como $X = X^+ \cup X^-$ donde $X^+ = \{X^t, X^f, X^m\}$ es un área de interés seleccionada arbitrariamente y X^- el conjunto de variables excluidas. El subconjunto X^t denota las variables “objetivo”, que son las que más nos interesa optimizar y actualizar; el subconjunto X^f denota las “frontera” variables de interés que son las que comparten al menos un factor con X^m ; el subconjunto X^m denota las variables “a marginalizar”, que serán las que efectivamente necesitarán ser contextualizadas con una topología adecuada de factores *prior* para mantener la consistencia de la incertidumbre. El número de variables y factores seleccionados determinará la complejidad de los procedimientos posteriores, como la optimización del grafo local.

En este trabajo se utiliza una política de selección por covisibilidad. A medida que el robot navega y se agregan nuevos estados y variables al grafo, el sistema SLAM realiza un seguimiento de un estado de referencia que representa la localización más actual del robot dentro del grafo; este estado de referencia se notará como \mathbf{x}_{ref} . El estado de referencia y las variables directamente relacionadas se considerarán las más significativas a optimizar, siguiendo esta idea, el algoritmo busca recuperar las variables circundantes y los factores, explotando la vecindad de los nodos. La Figura 2 ilustra una situación de ejemplo donde el grafo \mathcal{G} ha sido segmentado y se han definido los conjuntos de variables requeridos.

Es posible limitar la cantidad de variables en X^+ predefiniendo un cierto número N de estados. La complejidad del algoritmo estará determinada por la conectividad del área local y no por el tamaño del grafo completo. Además, un número acotado de variables y factores seleccionados determinará una complejidad acotada de los procedimientos posteriores, como la optimización del grafo local.

B. Prior Aproximado Utilizando Árboles de Chow-Liu

Dado que la matriz de covarianza aproximada $\tilde{\Sigma}$ se construye “por bloques” puede suceder que no contenga información de correlación entre algunas variables, y esto esto dificulta su utilización directa en un factor *prior* denso. Existirá información de correlación entre dos variables solo cuando estas fueran seleccionadas al mismo tiempo en una optimización local. Por este motivo, en vez de un único factor *prior* que correlacione los estados pertenecientes a X^m , se utiliza una colección de factores *priors* que proporcionen un conjunto de restricciones y contextualicen las variables X^m , y asegure la observabilidad de la “frontera” X^f .

La topología de factores *prior* utilizada es un árbol de Chow-Liu [15] dado que existe un algoritmo para construirlo

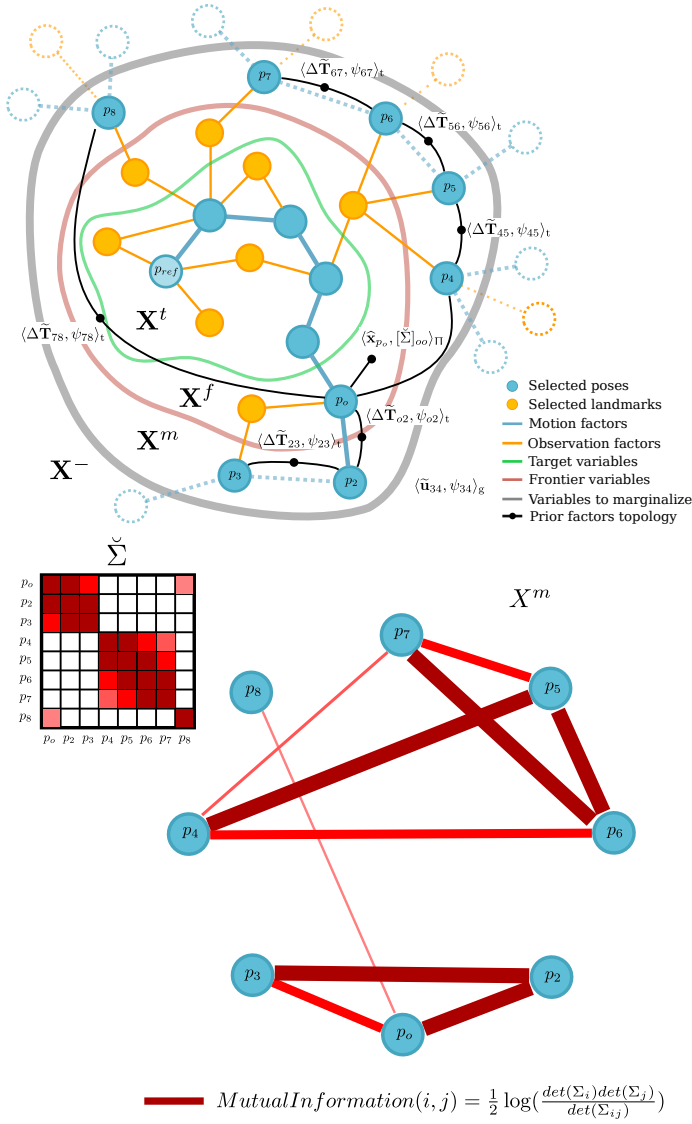


Fig. 2: Ejemplo de selección de área local y definición de la topología *prior*. Los segmentos en negro con punto representan factores *prior*s agregados pertenecientes a \mathcal{P} . $\langle \bullet, \bullet \rangle_t$ denota los parámetros integrados de cada factor, como las mediciones virtuales calculadas, su incertidumbre asociada y el modelo de función. La pose de origen p_o tiene un factor *prior* unario asociado y se impone una topología en árbol entre variables de X^m , ψ_{ij} denota una incertidumbre obtenida aplicando la función $\psi(\mathbf{x}_i, \mathbf{x}_j)$ como se explica en la ecuación (10). La imagen inferior muestra el estado de la covarianza aproximada al momento de definir la topología *prior* y su representación en grafo utilizando la información mutua como aristas entre las poses (Sec. 2).

de manera eficiente y garantiza aproximar la distribución original $\check{\Sigma}$. Como raíz del árbol se utiliza el estado menos incierto, es decir, el de menor volumen de covarianza $\sqrt{\det(\check{\Sigma}_{xx})}$, y sus aristas se determinan maximizando la información mutua (*mutual information*, MI) entre cada par de variables. Para lograr esto, primero se construye un grafo entre los estados de X^m con aristas pesadas de acuerdo a la *mutual information*, y luego se obtiene un árbol máximo utilizando el algoritmo de Kruskal con los pesos de las aristas multiplicados por -1 . La figura 2 muestra un escenario de ejemplo, donde puede

verse en la imagen superior la selección de área y como se relacionan las poses de los estados de X^m con la topología de factores *prior*, y en la imagen inferior el estado en que se encontraba la matriz de covarianza aproximada $\check{\Sigma}_{xx}$ y el grafo de *mutual information* entre los estados. Al estado raíz del árbol *prior*, al ser el menos incierto, se le agrega un *prior* unario que servirá como ancla para mantener la optimización local contextualizada en una determinada área global. Los estados del árbol se los relaciona utilizando factores relativos de incremento entre poses definidos como:

$$f_i(\mathbf{x}_i, \mathbf{x}_j) \propto \exp\left(-\frac{1}{2} \|\text{Log}(\Delta \tilde{\mathbf{T}}_{i,j}^{-1} \mathbf{T}_i^{-1} \mathbf{T}_j)\|_{\psi(\mathbf{x}_i, \mathbf{x}_j)}^2\right), \quad (9)$$

$\Delta \tilde{\mathbf{T}}_{i,j}^{-1} = \mathbf{t}(\hat{\mathbf{x}}_{k_i}, \hat{\mathbf{x}}_{k_j}) = \hat{\mathbf{T}}_i^{-1} \hat{\mathbf{T}}_j$ es una medición virtual de transformaciones relativas construidas con las estimaciones más recientes de las variables, \mathbf{t} refiere a la función de composición de transformaciones de pose, y

$$\psi(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{J} \times \mathbf{M} \times \text{SPD}\left(\begin{bmatrix} [\check{\Sigma}]_{ii} & [\check{\Sigma}]_{ij} \\ [\check{\Sigma}]_{ji} & [\check{\Sigma}]_{jj} \end{bmatrix}\right) \times \mathbf{M}^\top \times \mathbf{J}^\top, \quad (10)$$

con $\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{t}(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i} \big|_{\hat{\mathbf{x}}} & \frac{\partial \mathbf{t}(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_j} \big|_{\hat{\mathbf{x}}} \end{bmatrix}$ es la matriz jacobiana de las derivadas parciales de la función de composición con respecto a las variables, y

$$\mathbf{M} = \begin{bmatrix} \frac{\partial(\mathbf{x}_i \boxplus \Delta \mathbf{x}_i)}{\partial \Delta \mathbf{x}_i} \big|_{\hat{\mathbf{x}}, \Delta \mathbf{x}=0} & \mathbf{0} \\ \mathbf{0} & \frac{\partial(\mathbf{x}_j \boxplus \Delta \mathbf{x}_j)}{\partial \Delta \mathbf{x}_j} \big|_{\hat{\mathbf{x}}, \Delta \mathbf{x}=0} \end{bmatrix}, \quad (11)$$

las derivadas del operador \boxplus con respecto al incremento de la optimización (Sec. III). Como las poses no pertenecen a un espacio euclidiano, se aplica \mathbf{M} para recuperar la representación de covarianza en el espacio original de los estados. La función $\text{SPD}(\bullet)$ toma los bloques de covarianza cruzada disponibles y calcula la matriz semidefinida positiva simétrica más cercana en la norma de Frobenius utilizando una matemática cerrada derivada por Nicholas J. Higham¹ [22]. Esto es necesario ya que la estrategia de actualización por bloques (explicada con más detalle en la Sección IV-D) puede conducir a una covarianza aproximada $\check{\Sigma}$ que no es semidefinida positiva. De esta manera, la función ψ propaga la información de correlación entre poses disponibles en $\check{\Sigma}$ al espacio del modelo de transformaciones relativas \mathbf{t} usando una propagación de primer orden.

Hasta donde sabemos, no existe ningún otro sistema SLAM en la literatura que haya aplicado esta metodología y represente la maquinaria central que hace que el novedoso concepto de covarianza aproximada más dispersa sea útil.

C. Optimización del Grafo Local

El gráfico local se optimiza aplicando una optimización no lineal iterativa (como se explica en la Sección III) sobre el subconjunto seleccionado de X^+ variables y factores \mathcal{G}^+ , con

¹Un ejemplo de implementación se puede encontrar en <https://www.mathworks.com/matlabcentral/fileexchange/42885-nearestspd>

la topología *prior* incluida (como se muestra en la Figura 2). La matriz de hessiana \mathbf{H}^+ calculada durante la optimización invierte y la covarianza resultante $\Sigma^+ = (\mathbf{H}^+)^{-1}$ se considera una estimación de incertidumbre aproximadamente consistente para las variables en X^+ . La topología *prior* de factores virtuales modela una distribución marginal $p(X^m)$ que resume la incertidumbre global en las estimaciones obtenidas hasta ese momento.

D. Actualización del Estado

Luego de la optimización local, la estimación más reciente de la media de las variables es actualizada directamente, mientras que la covarianza local $\Sigma^+ = (\mathbf{H}^+)^{-1}$ se utiliza para actualizar los bloques correspondientes a las variables seleccionadas en la matriz de covarianza aproximada $\hat{\Sigma}$. No hay garantías de que la covarianza aproximada contenga información de correlación entre todas las variables del grafo. Esto es inherente a la optimización local, ya que no todos los estados y *landmarks* se incluyen y actualizan al mismo tiempo; aunque se garantiza que cada variable tenga calculada una covarianza marginal en el bloque diagonal.

En los casos en que el tipo de sensor genera observaciones de dimensión inferior a las de los *landmarks* (por ejemplo, cuando se utiliza una cámara monocular), la matriz hessiana resultante \mathbf{H}^+ puede ser singular y puede no ser directamente invertible. Para manejar esta situación, es un enfoque común aplicar una generalización de la inversa como la pseudoinversa de Moore-Penrose [23].

V. RESULTADOS Y CONCLUSIONES

El trabajo se enmarca en una línea de investigación sobre estimación y consistencia probabilística con resultados reportados en el artículo de conferencia Castro et al. [1], donde se pone a prueba la teoría en una implementación en Matlab utilizando únicamente una formulación de SLAM Visual, y utilizando una topología *prior* heurística en forma de cadena (*chain topology*). El método muestra resultados comparables con una optimización completa del grafo (*batch optimization*) y superando a otros enfoques canónicos. A partir de estos resultados se trabaja en extender el método incluyendo mediciones inerciales y realizar una implementación en C++ capaz de operar en tiempo real. El sistema se encuentra operacional y puede observarse en la figura 1 junto con un entorno de simulación de drones y *landmarks* detectables.

El método se encuentra funcionando con la nueva topología *prior* basada en árboles de Chow-Liu y se tienen planificadas evaluaciones de desempeño para reportar en *journals* internacionales. Al día de hoy, nos encontramos trabajando en mejorar la robustez de la implementación y su velocidad para realizar pruebas en datasets con datos reales. Es posible encontrar material suplementario de video del sistema funcionando en: https://drive.google.com/file/d/1_-MpZs7-6z7Gizgn2W0YD3iucDmWdz1Q

REFERENCIAS

- [1] G. Castro, F. Pessacq, and P. De Cristóforis, "Consistent slam using local optimization with virtual prior topologies," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 3321–3328.
- [2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard, "Simultaneous Localization And Mapping: Present, Future, and the Robust-Perception Age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1–139, 2016.
- [3] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT press, 2005.
- [4] F. Dellaert and M. Kaess, "Factor Graphs for Robot Perception," *Foundations and Trends in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.
- [5] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.
- [6] F. Bai, T. Vidal-Calleja, and G. Grisetti, "Sparse pose graph optimization in cycle space," *IEEE Transactions on Robotics*, 2021.
- [7] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proceedings of the 1997 American Control Conference (Cat. No. 97CH36041)*, vol. 4. IEEE, 1997, pp. 2369–2373.
- [8] V. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 422–429, 2019.
- [9] T. Qin, S. Cao, J. Pan, and S. Shen, "A General Optimization-based Framework for Global Pose Estimation with Multiple Sensors," in *Computing Research Repository*, 01 2019.
- [10] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.
- [11] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [12] G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing," *Journal of Field Robotics*, vol. 27, no. 5, pp. 587–608, 2010.
- [13] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, March 2015.
- [14] R. Mur-Artal and J. D. Tardós, "Visual-Inertial Monocular SLAM With Map Reuse," *IEEE Robotics and Automation Letters*, April 2017.
- [15] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.
- [16] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2O: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*, May 2011.
- [17] S. Agarwal and K. Mierle, "Ceres solver," <http://ceres-solver.org>.
- [18] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree," *Intl. J. of Robotics Research, IJRR*, vol. 31, no. 2, pp. 217–236, February 2012.
- [19] J. Solà, J. Deray, and A. Dinesh, "A micro Lie theory for state estimation in robotics," *Technical Report IRI-TR-18-01, CSIC-UPC*, 2018. [Online]. Available: <https://www.iri.upc.edu/publications/show/2089>
- [20] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-Manifold Preintegration for Real-Time Visual-Inertial Odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.
- [21] F. Dellaert and G. Contributors, "borglab/gtsam," May 2022. [Online]. Available: <https://github.com/borglab/gtsam>
- [22] N. J. Higham, "Computing a nearest symmetric positive semidefinite matrix," *Linear algebra and its applications*, pp. 103–118, 1988.
- [23] R. Penrose, "A generalized inverse for matrices," in *Mathematical proceedings of the Cambridge philosophical society*, vol. 51, no. 3. Cambridge University Press, 1955, pp. 406–413.