

TABLA DE CONTENIDOS

ESTADO DEL ARTE.....	1
MÉTODOS USADOS.....	2
ORB-SLAM.....	2
SVO-SLAM.....	2
DATASET.....	3
EXPERIMENTOS.....	4
RESULTADOS.....	5
COMPARATIVA ORB-SLAM2 Y ORB-SLAM 3.....	5
COMPARATIVA SVO-SLAM VIO E IMU.....	7
COMPARATIVA COMPLETA.....	9
BIBLIOGRAFÍA.....	10
ANEXO.....	11

ESTADO DEL ARTE

Este proyecto tiene como objetivo la comparación de distintos métodos de SLAM visual. Los cuales pertenecen a una familia que deriva de las técnicas SLAM.

Las técnicas SLAM (*Simultaneous Localization And Mapping*) son un conjunto de técnicas utilizadas actualmente en muchos ámbitos de la robótica y, en específico, de vehículos autónomos, cuyo objetivo principal es la estimación de la pose (localización) y la recreación del entorno (mapeo) del vehículo de manera simultánea. Mientras que este se va desplazando, ya sea con el objetivo de explorar el entorno o como objetivo secundario para realizar otra tarea principal.

Por la naturaleza de estas técnicas, están orientadas a vehículos móviles, que se deben de apoyar en varios sensores y actuadores. Al unir todos estos factores, existen múltiples posibles puntos de error. Empezando por la utilización de robots móviles, que ha representado y sigue representando un gran desafío para la ingeniería (se ha estado estudiando este área desde hace décadas para solucionarlo). Y continuando por la cantidad de sensores (en número y tipos) que se deben utilizar y todo el ruido que aportan. Lo que afecta en gran medida a la precisión y exactitud de las estimaciones que puedan realizar estos métodos SLAM. Además, esta complejidad aumenta, inevitablemente, por la dificultad representativa de los entornos, que pueden ser cambiantes y dinámicos. Otro problema de complejidad, se encuentra en el aspecto computacional, ya que los métodos SLAM pueden consumir una gran capacidad de cómputo, a lo que se tiene que afrontar con los recursos a bordo de la máquina.

Por todo esto, se tiene la necesidad de trabajar con cantidades enormes de información (contaminada por ruido) y; por tanto, las soluciones más exitosas han sido las basadas en técnicas probabilísticas. Éstas técnicas pretenden hacer frente a todas las fuentes de incertidumbre involucradas, aplicando (en gran medida) el filtro de Kalman (o alguno de sus derivados) u otras técnicas basadas en el teorema de Bayes o mapas de ocupación de celdillas.

Dentro de esta familia, el SLAM visual (VSLAM) es un conjunto de métodos SLAM que se basan en el empleo de imágenes (obtenida mediante cámaras monoculares, stereo o RGB-D) para desarrollar el mapeado y localización simultáneos. El primer proyecto de SLAM visual se realizó con una cámara monocular y, posteriormente, fueron haciéndose mejoras hasta conseguir desarrollar la técnica con cámaras estéreo y RGB-D.

Además de los grandes avances y prestaciones, también existen limitaciones para que el uso de esta técnica sea correcto. Algunas de ellas son que se pueden mapear los entornos en movimiento sólo si éstos no dominan la vista, y que los cambios de iluminación tienen que ser escasos.

Los algoritmos de VSLAM tienen 5 pasos diferenciados:

1. Inicialización
2. Localización
3. Mapeado
4. Re-localización
5. Optimización del mapeado.

Hoy en día hay muchísimos métodos de SLAM visual, como ORB-SLAM, PTAM, OpenVSLAM, SVO-SLAM o SVDL entre otros.

MÉTODOS USADOS

Concretamente, durante este proyecto se pretende comparar dos métodos de VSLAM, y algunas opciones que permiten estos métodos. En primer lugar, compararemos las dos últimas versiones de ORB-SLAM: ORB-SLAM2 y ORB-SLAM3; para poder ver si existen mejoras entre ambas. Y por otro lado, se utilizará SVO-SLAM, el cual permite utilizar VIO ('Visual Inertial Odometry') o un sensor IMU. Éstos han sido los escogidos debido a sus buenos resultados y características .

Para poder entender mejor las diferencias que aportan cada uno de los métodos y las opciones, veremos en qué se basan cada uno de los métodos.

ORB-SLAM

El método de ORB-SLAM es un método de VSLAM, desarrollado en la Universidad de Zaragoza. El cuál, se lanzó en principio únicamente con la posibilidad de utilizar cámaras monoculares. Aunque, para las dos versiones posteriores (las que vamos a comparar) permiten, además, el uso de cámaras stereo y RGB-D.

Este modelo es uno de los más conocidos, debido a los buenos resultados que aporta. Esto es gracias a la forma de capturar los features que utiliza (features ORB), los cuáles dan nombre al método. Los features ORB (Oriented FAST and Rotated BRIEF) se centran en mezclar y mejorar diferentes features preexistentes. Mejorando la velocidad de los features SIFT y FAST. Además de mejorar los BRIEF ante rotaciones y ruidos.

Por un lado, ORB-SLAM2 mejora la versión original, posibilitando el uso de cámaras stereo y RGB-D. Para estas cámaras se permite el uso de lentes pinhole. Además, también añade el cierre de bucle y la reutilización del mapa.

Mientras que ORB-SLAM3 mantiene estas características, también mejora las anteriores añadiendo diversos elementos. En primer lugar, permite las lentes fisheye y da un gran avance habilitando la odometría visual. Además de aumentar la precisión de los resultados, permite que existan periodos con baja información visual sin perturbaciones notables, es decir, sin un gran aumento del error.

Ambas técnicas están publicadas en repositorios de GitHub para poder utilizarlo de diferentes formas, aunque en los experimentos posteriores, utilizaremos los paquetes para ROS.

SVO-SLAM

El segundo método a comparar proviene de la Universidad de Zurich, y aunque utiliza también features ORB, se basa en una serie de técnicas denominadas como semidirectas. Estas técnicas fusionan las dos ramas tradicionales de odometría visual utilizada en la visión por computador.

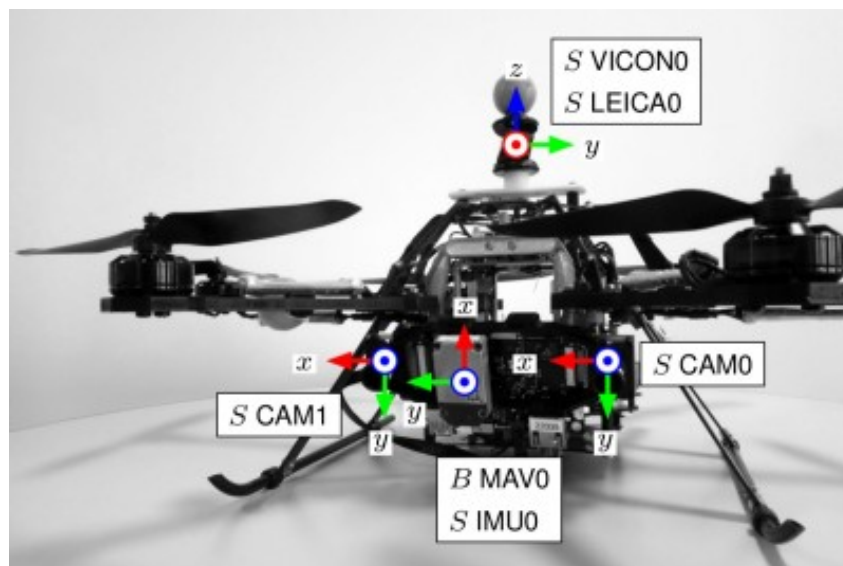
Por un lado están las técnicas basadas en features (como los ORB) en los que se pretenden buscar puntos característicos en cada uno de los frames. Esto, aunque da buenos resultados, computacionalmente es muy costoso.

Por el otro, tenemos los métodos directos, que no buscan puntos característicos, sino que se sirven del cambio de intensidades en los píxeles para intentar extrapolar lo ocurrido en la escena. Esta presenta mejores tiempos de cómputos, pero no da tan buenos resultados. Para intentar unificar los puntos fuertes de ambos métodos, se desarrollan los métodos semidirectos de odometría visual, lo que da nombre a este método de SLAM (SVO, Semi Visual-Odometry). Estos métodos se basan en buscar features únicamente en algunos frames (aligerando la carga computacional de estos métodos), y para estimar la posición de estos en los frames en los que no se detectan, se utilizan métodos directos. De esta manera se busca equilibrar los mejores puntos de ambas técnicas.

Este método está publicado en un repositorio de GitHub, el cual nos permite utilizar dos métodos para el cálculo de la pose: VIO (Visual Inertial Odometry) y el sensor IMU. Los que se van a comparar. Ambas opciones están disponibles para ROS.

DATASET

Existen muchos dataset disponibles para los experimentos que se quieren realizar. En un primer momento, pensamos en utilizar uno de los más conocidos y de mejores prestaciones, KITTI. Debido a que es muy reconocido, y creíamos que no daría muchos problemas. Además, los modelos de ORB-SLAM2 y ORB-SLAM3 están probados con este dataset, dando muy buenos resultados. Sin embargo, al probar la compatibilidad con el paquete de ROS del SVO-SLAM, tuvimos problemas para ponerlo en marcha. Por lo cual, tras mirar otros dataset existentes, decidimos utilizar el dataset de EuRoC, el cual nos daba los bag para ROS directamente y no nos daba problemas al ejecutarlo con ninguno de los paquetes.



EuRoC es un dataset realizado por el Instituto Tecnológico de Zurich, que nos facilita diferentes datos de varias trayectorias de un UAV micro, al cual denominan MAV.

Las trayectorias proporcionadas están realizadas en 2 escenarios diferentes (MACHINE HALL y VICON ROOM), cada uno obtiene el Ground-truth mediante un sensor diferente (LEICA y VICON, respectivamente). Además, de cada escenario podemos elegir diferentes dificultades dependiendo del número de features capturados.

El MAV, comentado anteriormente, nos da toda la información que necesitamos gracias a varios sensores que posee:

- **Cámara Stereo monocolor Aptina MT9V034**, la cual nos servirá tanto para los modos estéreo y mono (utilizando sólo una de las cámaras) de los paquetes SLAM.
- **MEMS IMU ADIS1644**, que nos servirá para el SVO-SLAM con IMU.
- **VICON MOTION CAPTURE SYSTEM**, el cual nos proporcionará el ground-truth de la pose en las trayectorias realizadas en el *Vicon Room*.
- **LEICA MS50 LASER TRACK**, que nos facilitará el ground-truth de la posición en el *Machine Hall*.

Todos los datos que nos proporcionan estos sensores están temporalmente alineados, los que nos resultará muy útil para la comparación de los resultados.

Sin embargo, dentro del dataset no hemos encontrado los datos que necesitamos para poder transformar las poses obtenidas a la del ground-truth. Por lo que al realizar el procesamiento de los datos obtenidos, utilizamos métodos alternativos para compararlos. Debido a esto, los datos obtenidos no se han podido ajustar de la mejor manera de la que lo hemos hecho al ground truth para la comparativa.

EXPERIMENTOS

Para poder hacer una comparativa de estos métodos, hemos hecho un total de dieciséis configuraciones distintas, las cuales combinan los distintos modelos ORB-SLAM2, ORB-SLAM3, SVO-SLAM con VIO y SVO-SLAM con IMU con dos escenarios posibles. De forma adicional, se utilizan dos tipos de cámaras en los experimentos: estéreo y monocular. Para los escenarios, hemos seleccionado dos trayectorias ofrecidas en EuRoC, cada una en cada uno de los escenarios previamente explicados, para tener mayor diversidad.

Debido a los requisitos de los paquetes, nos dividimos los experimentos de la siguientes forma, consiguiendo paralelizar las tareas::

- Por un lado, Pedro se encarga de realizar los experimentos de ambos paquetes de ORB-SLAM. Ya que ambos paquetes están pensados y preparados para ejecutarse en el mismo sistema operativo (Ubuntu 18.04).
- Mientras que por el otro lado, Francisco tenía como tarea realizar los experimentos del SVO-SLAM, tanto con IMU como con VIO. Debido a que ambas opciones están en el mismo paquete, está preparado para Ubuntu 20.04.

Tras obtener los resultados, se le realiza un procesamiento para poder efectuar la comparación. Con los resultados de cada uno de los experimentos, primero se extraen los datos de los ficheros donde están guardados. Asociándolos a las marcas de tiempo correspondiente. A lo que le sigue una interpolación, para que todas las señales tengan el mismo número de muestras.

El único paso que faltaría, sería la transformación para ver todas las poses bajo el mismo sistema de referencia. Sin embargo, como se dijo anteriormente, en el dataset de EuRoC no

hemos podido encontrar la relación entre el sistema de las cámaras, el de la IMU y el de los sensores que proporcionan el ground-truth. Por lo cual, se hizo de manera manual.

Para la traslación, se decidió restar el punto de partida de cada secuencia de pose, de tal manera que todas las secuencias de poses de los sensores empiezan en el origen.

Mientras que, para las rotaciones de los distintos ejes, nos basamos en esquemas dados por el dataset para obtener los ángulos de giro. Además de realizar un ajuste fino de forma manual para poder obtener los menores errores posibles.

Por esto entendemos que los errores que obtengamos no serán los menores errores posibles. Sin embargo, sí que seguirán la misma tendencia.

RESULTADOS

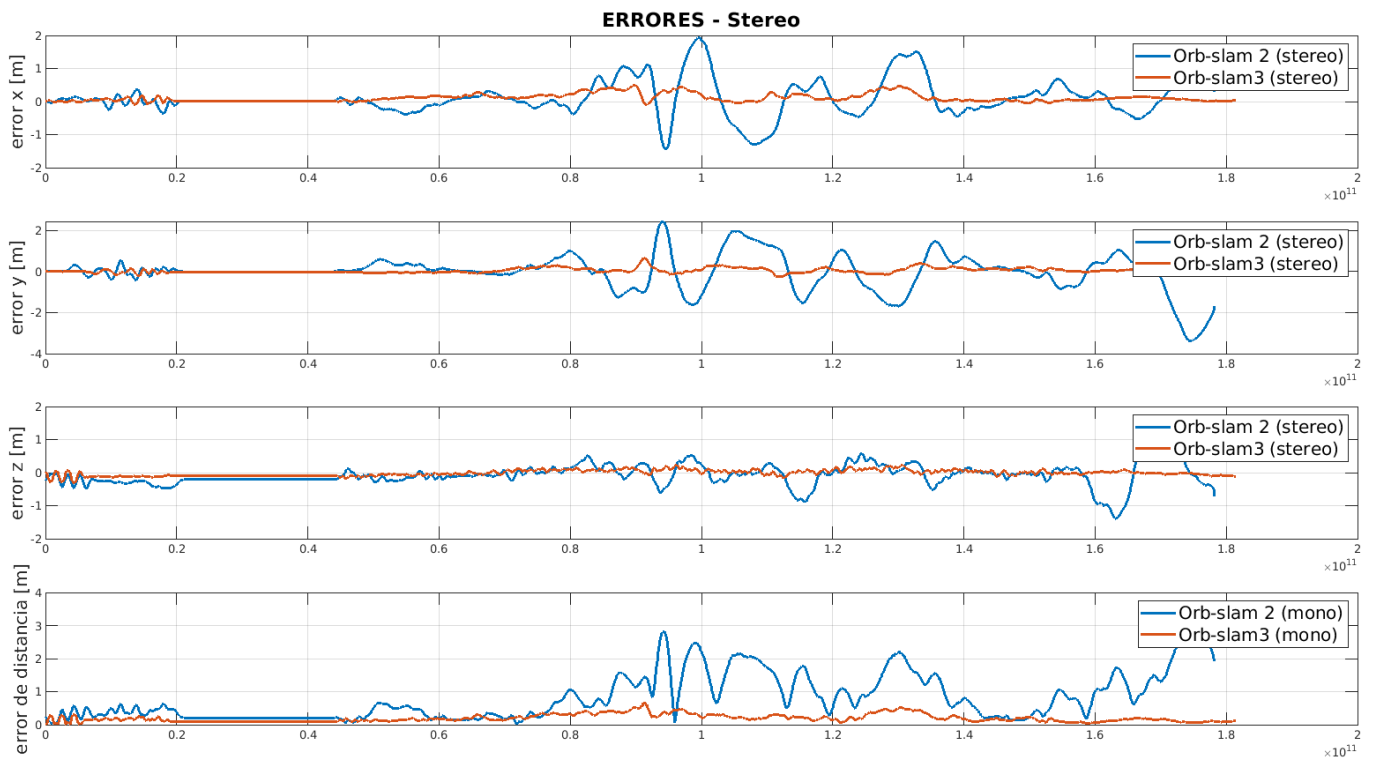
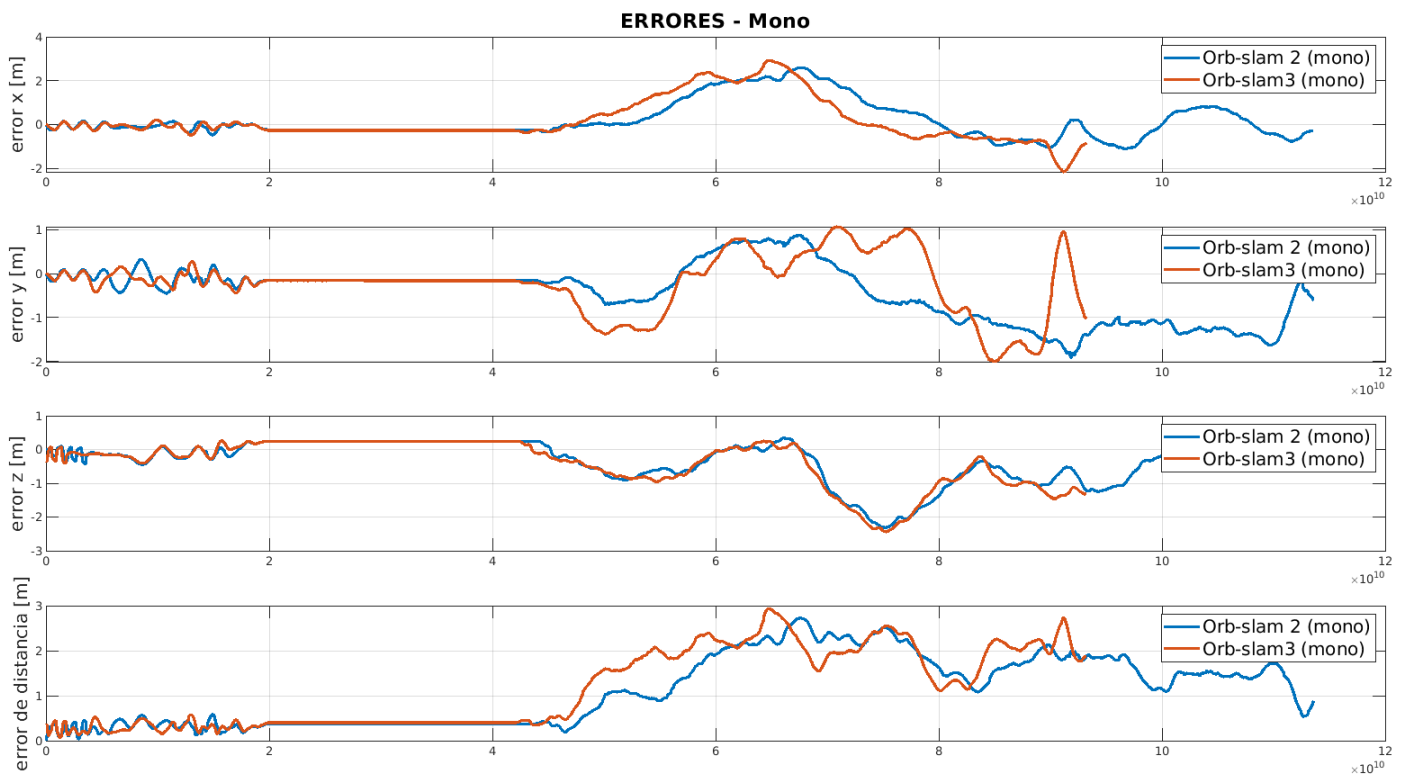
COMPARATIVA ORB-SLAM2 Y ORB-SLAM 3

Para poder comparar estos dos modelos, hemos hecho ocho experimentos. Ambos modelos bajo cámaras monoculares y estéreo bajo dos dataset distintos (a los que denominaremos 'MH 01' para la Machine Hall y 'V1-02', para la habitación Vicon). Con lo cual, hemos podido obtener las siguientes gráficas. Además, también hemos podido extraer los errores máximo, medio y cuadrático medio de los experimentos. Vamos a ver los más relevantes.

ERROR MÁXIMO [m]	<i>STEREO MH 01</i>	<i>STEREO V1-02</i>
<i>ORB-SLAM2</i>	3.4615	0.4227
<i>ORB-SLAM3</i>	0.6707	0.3799

ERROR MEDIO [m]	<i>STEREO MH 01</i>	<i>STEREO V1-02</i>
<i>ORB-SLAM2</i>	0.6707	0.1344
<i>ORB-SLAM3</i>	0.1957	0.1293

ERROR CUADRÁTICO MEDIO [m]	<i>STEREO MH 01</i>	<i>STEREO V1-02</i>
<i>ORB-SLAM2</i>	0.8243	0.2759
<i>ORB-SLAM3</i>	0.4257	0.2749

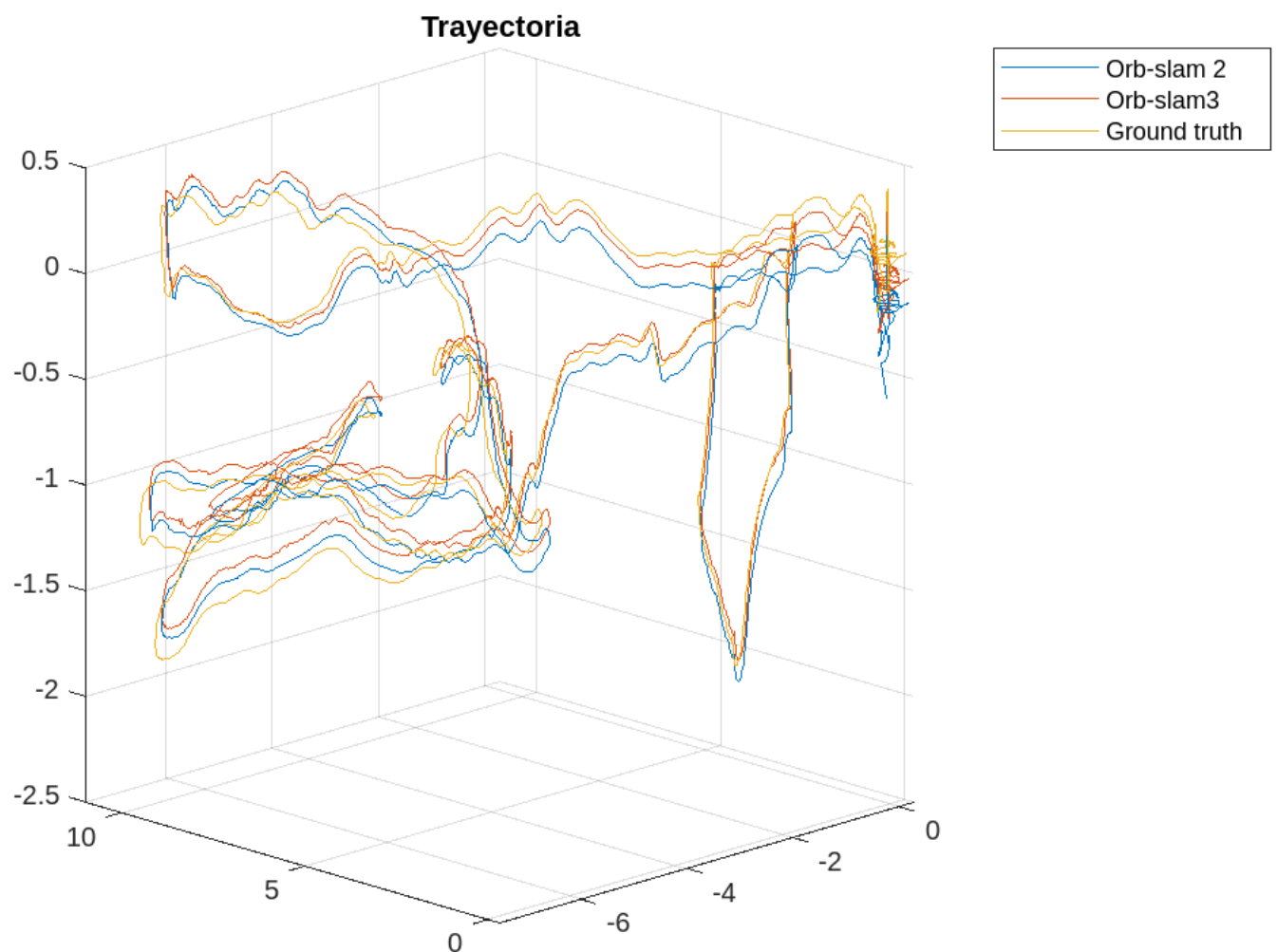


Ante estas gráficas y esta tabla, podemos ver dos tendencias claras. En primer lugar, se ve que ambos métodos mejoran al utilizar cámaras stereo en vez de monoculares.

La segunda tendencia se ve más claramente en la modalidad stereo (que es el mejor modo de ORB-SLAM2 y 3). Y se ve que el modelo de ORB-SLAM3 mejora en gran medida los errores de la versión anterior.

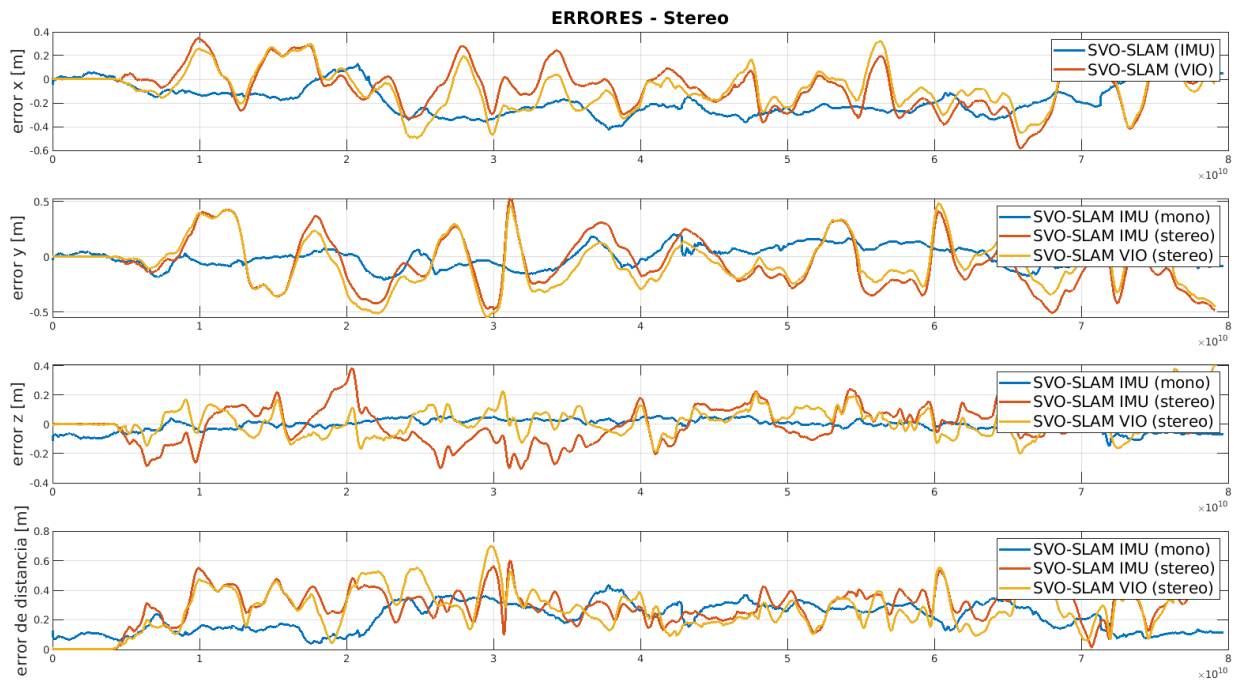
De esto, podemos concluir, que de estas modalidades, la mejor sería ORB-SLAM3 conjuntamente a una cámara stereo.

Como última gráfica de esta comparativa, tenemos la trayectoria en el espacio de ambos métodos respecto al ground truth (ambas en stereo).



COMPARATIVA SVO-SLAM VIO E IMU

Hemos realizado los mismos experimentos, con el paquete de SVO-SLAM, también utilizando cámaras monoculares y stereo. Además de los mismos escenarios. En este caso, vamos a comparar el mismo paquete, pero utilizando la IMU o la VIO.



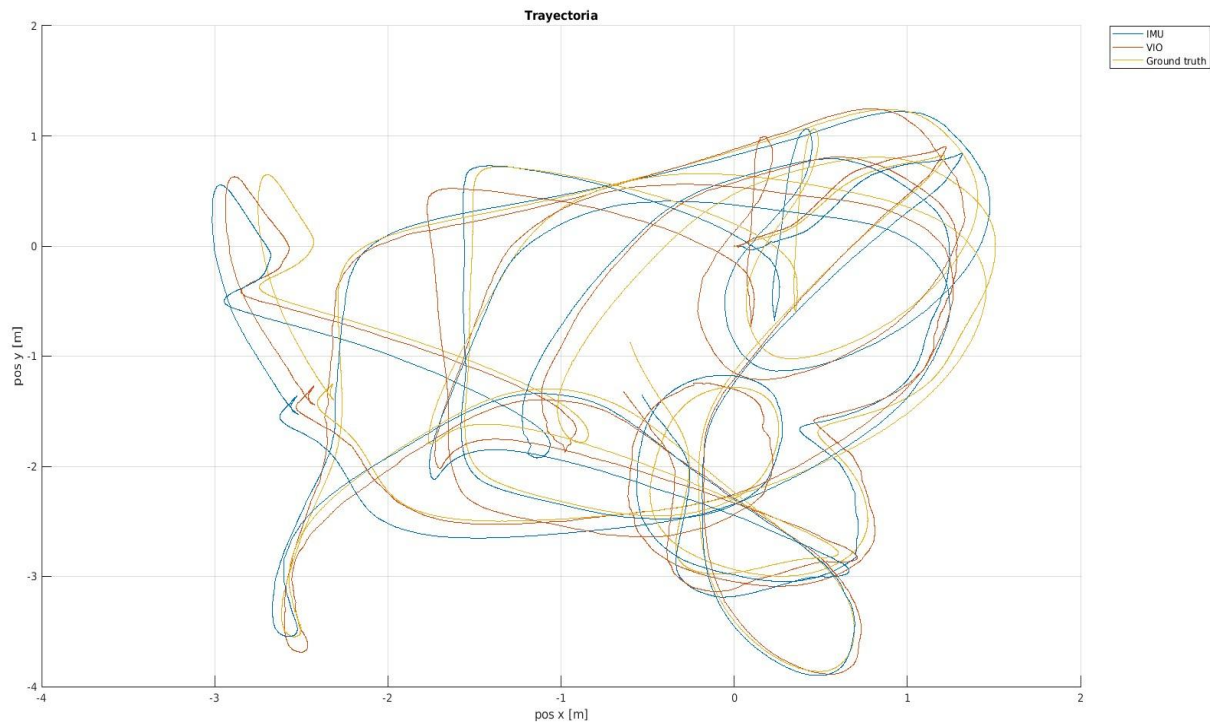
En este, únicamente de las gráficas no se puede sacar una conclusión tan clara como en el caso anterior. Por lo tanto, para obtener alguna idea en claro, debemos utilizar las tablas con los errores (como en el caso anterior).

ERROR MÁXIMO [m]	<i>STEREO MH 01</i>	<i>MONO MH 01</i>	<i>STEREO V1-02</i>
<i>SVO-SLAM IMU</i>	1.1698	2.7954	0.6232
<i>SVO-SLAM VIO</i>	1.8513	1.8498	0.6983

ERROR MEDIO [m]	<i>STEREO MH 01</i>	<i>MONO MH 01</i>	<i>STEREO V1-02</i>
<i>SVO-SLAM IMU</i>	0.4794	0.8782	0.2974
<i>SVO-SLAM VIO</i>	0.5215	0.3740	0.2735

ERROR CUADRÁTICO [m]	<i>STEREO MH 01</i>	<i>MONO MH 01</i>	<i>STEREO V1-02</i>
<i>SVO-SLAM IMU</i>	0.6693	0.8826	0.5223
<i>SVO-SLAM VIO</i>	0.6945	0.5594	0.4971

Ante estos datos, se puede observar que la diferencia no es tan grande como entre ORB-SLAM 2 y 3. Sin embargo, sí hay cierta tendencia, que se corresponde con los datos proporcionados por el repositorio oficial, que se cumple. Los casos en los que se utiliza el VIO da cierta mejora ante la IMU. A continuación, se muestra la trayectoria del experimento, con el ground-truth y ambos métodos.



COMPARATIVA COMPLETA

Finalmente, tenemos que dentro de los métodos ORB-SLAM, el mejor de ellos es la última versión (ORB-SLAM3). A la vez, aunque dentro del método de SVO-SLAM no hay una diferencia apreciable, podemos comparar ORB-SLAM3 con SVO-SLAM con VIO.

De entre estos dos, podemos tomar los mejores resultados obtenidos. De lo que se puede observar que el mejor error lo da ORB-SLAM3. Habiendo una diferencia importante entre ambos.

STEREO V1-02	<i>ERROR MÁXIMO</i>	<i>ERROR MEDIO</i>	<i>ERROR CUADRÁTICO MEDIO</i>
ORB-SLAM3	0.3798	0.1283	0.2740
SVO-VIO	0.6983	0.2735	0.4971

BIBLIOGRAFÍA

- Campos, C., Elvira, R., Rodriguez, J. J. G., M. Montiel, J. M., & D. Tardos, J. (2021). ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM. *IEEE Transactions on Robotics: A Publication of the IEEE Robotics and Automation Society*, 37(6), 1874–1890. <https://doi.org/10.1109/tro.2021.3075644>
- Forster, C., Zhang, Z., Gassner, M., Werlberger, M., & Scaramuzza, D. (2017). SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems. *IEEE Transactions on Robotics: A Publication of the IEEE Robotics and Automation Society*, 33(2), 249–265. <https://doi.org/10.1109/tro.2016.2623335>
- Kmavvisualinertialdatasets – ASL datasets*. (n.d.). Ethz.Ch. Retrieved May 28, 2023, from <https://projects.asl.ethz.ch/datasets/doku.php?id=kmavvisualinertialdatasets>
- Mur-Artal, R. (n.d.). *ORB_SLAM2: Real-Time SLAM for Monocular, Stereo and RGB-D Cameras, with Loop Detection and Relocalization Capabilities*.
- Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics: A Publication of the IEEE Robotics and Automation Society*, 31(5), 1147–1163. <https://doi.org/10.1109/tro.2015.2463671>
- Mur-Artal, R., & Tardos, J. D. (2017). ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics: A Publication of the IEEE Robotics and Automation Society*, 33(5), 1255–1262. <https://doi.org/10.1109/tro.2017.2705103>

ORB_SLAM3: ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM. (n.d.).

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *2011 International Conference on Computer Vision*.

SVO Pro Semi-direct Visual-Inertial Odometry and SLAM for Monocular, Stereo, and wide angle cameras. (n.d.). Uzh.ch. Retrieved May 28, 2023, from https://rpg.ifi.uzh.ch/svo_pro.html

ANEXO

Junto a esta memoria, se entregan 2 ficheros. En primer lugar, un fichero pdf con la presentación realizada el 26 de Mayo de 2023. Y un fichero zip, con todos los datos extraídos de los experimentos y un fichero personalizado para obtener cada una de las gráficas y el procesamiento realizado sobre los datos de cada uno de los experimentos.

De estos, tenemos los ficheros *.txt (los datos extraídos de cada experimento) y distintos ficheros *.m (uno para cada experimento). El experimento al que se refiere viene especificado en el nombre de cada fichero.