



UNIVERSIDAD  
DE MÁLAGA

Doctoral Dissertation

# Robust Visual SLAM in Challenging Environments with Low-texture and Dynamic Illumination


Rubén Gómez Ojeda  
2020

Tesis doctoral  
Ingeniería Mecatrónica  
Dpt. de Ingeniería de Sistemas y Automática  
Universidad de Málaga



UNIVERSIDAD  
DE MÁLAGA

AUTOR: Rubén Gómez Ojeda

 <http://orcid.org/00000-0002-5338-1746>

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional:

<http://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>

Cualquier parte de esta obra se puede reproducir sin autorización pero con el reconocimiento y atribución de los autores.

No se puede hacer uso comercial de la obra y no se puede alterar, transformar o hacer obras derivadas.

Esta Tesis Doctoral está depositada en el Repositorio Institucional de la Universidad de Málaga (RIUMA): [riuma.uma.es](http://riuma.uma.es)

UNIVERSIDAD DE MÁLAGA  
DEPARTAMENTO DE  
INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

El Dr. D. Javier González Jiménez, director de la tesis titulada "Robust Visual SLAM in Challenging Environments with Low-texture and Dynamic Illuminations" realizada por D. Rubén Gómez Ojeda, certifican su idoneidad para la obtención del título de Doctor en Ingeniería Mecatrónica.

Málaga, 26 de febrero de 2020

---

Dr. D. Javier González Jiménez

Dept. of System Engineering and Automation  
University of Málaga  
Studies in Mechatronics



# Robust Visual SLAM in Challenging Environments with Low-texture and Dynamic Illumination

AUTHOR: Rubén Gómez Ojeda

SUPERVISOR: Javier González Jiménez

Thesis defended on 26th February 2020

JURY:

Anthony Mandow (University of Málaga, Spain)

Javier Civera (University of Zaragoza, Spain)

Stefan Leutenegger (Imperial College of London, United Kingdom)



*A mis padres.*



# Table of Contents

<b>Table of Contents</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgments</b>	<b>v</b>
<b>Resumen de la Tesis</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.A Motivation . . . . .	2
1.B Contributions . . . . .	3
1.B.1 Contributions to SLAM in Low-textured Environments	4
1.B.2 Contributions to SLAM under Dynamic Illumination and HDR Environments . . . . .	5
1.B.3 Publications . . . . .	6
1.C Framework and Timeline . . . . .	7
1.D Outline . . . . .	9
<b>2 Simultaneous Localization and Mapping</b>	<b>13</b>
2.A Introduction . . . . .	13
2.B Overview of Visual SLAM Techniques . . . . .	16
2.B.1 Indirect Methods . . . . .	17
2.B.2 Direct Methods . . . . .	18
2.B.3 Semi-direct Approach . . . . .	18
2.C Remaining Challenges for Visual Odometry and SLAM . . . . .	19



<b>3</b>	<b>Robustness to Low-textured Environments</b>	<b>21</b>
3.A	Introduction . . . . .	21
3.B	Contributions . . . . .	22
3.C	Robust Stereo Visual Odometry through a Probabilistic Combination of Points and Line Segments . . . . .	23
3.D	Accurate Stereo Visual Odometry with Gamma Distributions .	36
3.E	PL-SVO: Semi-Direct Monocular Visual Odometry by Combining Points and Line Segments . . . . .	50
3.F	PL-SLAM: a Stereo SLAM System through the Combination of Points and Line Segments . . . . .	63
<b>4</b>	<b>Dealing with Dynamic Illumination and HDR Environments</b>	<b>91</b>
4.A	Introduction . . . . .	91
4.B	Contributions . . . . .	92
4.C	Learning-based Image Enhancement for Visual Odometry in Challenging HDR Environments . . . . .	93
4.D	Geometric-based Line Segment Tracking for HDR Stereo Sequences . . . . .	107
<b>5</b>	<b>Conclusions</b>	<b>121</b>
	<b>Bibliography</b>	<b>127</b>

## Abstract

In the last years, visual *Simultaneous Localization and Mapping* (SLAM) has played a role of capital importance in rapid technological advances, *e.g. mobile robotics* and applications such as *virtual, augmented, or mixed reality* (VR/AR/MR), as a vital part of their processing pipelines. As its name indicates, it comprises the estimation of the *state* of a robot (typically the *pose*) while, simultaneously, incrementally building and refining a consistent representation of the *environment*, *i.e.* the so-called *map*, based on the equipped sensors.

Despite the maturity reached by state-of-art visual SLAM techniques in controlled environments, there are still many open challenges to address before reaching a SLAM system robust to *long-term* operations in uncontrolled scenarios, where classical assumptions, such as static environments, do not hold anymore. This thesis contributes to improve robustness of visual SLAM in harsh or difficult environments, in particular:

**Low-textured Environments**, where traditional approaches suffer from an accuracy impoverishment and, occasionally, the absolute failure of the system. Fortunately, many of such low-textured environments contain planar elements that are rich in linear shapes, so an alternative feature choice such as *line segments* would exploit information from structured parts of the scene. This set of contributions exploits both type of features, *i.e.* points and line segments, to produce visual odometry and SLAM algorithms robust in a broader variety of environments, hence leveraging them at all instances of the related processes: *monocular depth estimation, visual odometry, keyframe selection, bundle adjustment, loop closing*, etc. Additionally, an *open-source C++ implementation* of the proposed algorithms has been released along with

the published articles and some extra multimedia material for the benefit of the community.

**Robustness to Dynamic Illumination** conditions is also one of the main open challenges in visual odometry and SLAM, *e.g.* *high dynamic range* (HDR) environments. The main difficulties in these situations come from both the limitations of the sensors, for instance automatic settings of a camera might not react fast enough to properly record dynamic illumination changes, and also from limitations in the algorithms, *e.g.* the track of interest points is typically based on *brightness constancy*. The work of this thesis contributes to mitigate these phenomena from two different perspectives. The first one addresses this problem from a *deep learning* perspective by enhancing images to invariant and richer representations for VO and SLAM, benefiting from the *generalization* properties of deep neural networks. In this work it is also demonstrated how the insertion of *long short term memory* (LSTM) allows us to obtain temporally consistent sequences, since the estimation depends on previous states. Secondly, a more traditional perspective is exploited to contribute with a purely *geometric-based* tracking of line segments in challenging stereo streams with complex or varying illumination, since they are intrinsically more informative.

## Acknowledgments

First and foremost, I would like to thank to my supervisor Prof. Javier González for spotting me in the Computer Vision lessons and offering me the opportunity of becoming a PhD student under his supervision. He wisely guided and inspired me through all of the stages in a PhD, putting me in the right direction at first, and giving valuable advise, support, and freedom to chase my own ideas in the second part. This work was only possible thanks to all of the fun discussions we had, that luckily converged to this thesis.

Of course, this has also been possible thanks to the amazing working environment in our laboratory at the University of Málaga, where I joined an extremely fun and supportive work group, that soon became my family. I will always remember how Javi, Raúl, or Curro, helped me in a complex onboarding to this world. I am also very grateful to Manu, for teaching me all the deep-learning I know, to Paco for persistently saving some of my abandoned works, and Jesús, we had a lot of fun in our constant collaborations and discussions, and in our crazy deadlines together. Undoubtedly, there is a special place for Mariano, for always putting me in the right direction and being the role-model I needed during my whole PhD, and Curro, for our uncountable nights discussing about robotics until early in the morning.

I would also like to thank Prof. Davide Scaramuzza, and the amazing RPG group (Guillermo, Henri, Zichao, Titus, Tammy), for hosting me as a visiting PhD student, and providing me a different and inspiring perspective that, unexpectedly, opened me a broader path in this community. Also, to the people from Oculus Zurich I had the opportunity to collaborate with during my internship, Christian, Luc, Michael, Alejo and specially Matia, for showing me the other side of this world, which in turn brought me back to Zurich more than a year ago.

I would like to thank my family and friends for being there. The necessary work to achieve a PhD does not only require technical skills, but also a key factor is the emotional support to help you evade from the hard moments we all suffer during this long period of the PhD life. This has been an extremely important part of this long journey that, hopefully, ends today.

Finally, I would like to thank Guillermo Gallego, Pablo Alcantarilla, and Alejo Concha for their valuable comments and reviews about this manuscript, and to the members of the Jury, Anthony Mandow, Javier Civera, and Stefan Leutenegger for helping in the evaluation of this Doctoral Thesis.

Rubén Gómez Ojeda  
Málaga  
February 2020

## Introducción

*Imagine* por un momento que la gente pudiera llevar dispositivos de *realidad virtual* (del inglés *VR*) que le permitieran unirse a un mundo con posibilidades ilimitadas, donde por ejemplo un ingeniero pudiera tener retroalimentación visual del modelo que se está diseñando, un cirujano pudiera ensayar sus siguientes cirugías con un *modelo 3D* realista del paciente, o una persona pudiera simplemente caminar por una ciudad y presenciar la recreación de algunos momentos históricos de la ciudad. Sin ir más lejos, imagínese que un *coche autónomo* podría llevarle al trabajo y recoger sus víveres momentos antes de volver a casa, o que los *vehículos autónomos no tripulados* (del inglés *UAV*) desempeñarían un papel vital en la asistencia en caso de catástrofe, o que los *robots inteligentes de telepresencia* podrían ayudar a las personas mayores en sus tareas cotidianas.

No hace tanto tiempo estas concepciones eran consideradas como ideas poco realistas o futuristas, únicamente dignas de historias de ciencia ficción como en la Figura 1 pero, de hecho, una mayoría de éstas son una realidad hoy en día, o al menos no sería atrevido asumir que lo serán en los próximos años. De hecho, existen algunas empresas de VR, como Oculus VR [5], que permiten a la gente sumergirse en películas en 3D y videojuegos, o caminar e interactuar por reconstrucciones virtuales de ciudades como Google Earth [2]. Así mismo, hay un número de compañías, como Waymo [8] (anteriormente conocido como el proyecto de conducción autónoma Google, Tesla [7] or Nuro [4], presentando sus diferentes implementaciones de autos proyectos que actualmente están siendo probados probando en ciudades como San Francisco.



**Figure 1:** Hoy en día no es atrevido imaginar un futuro en el que se pueda llevar algún equipo de realidad virtual transportándote a un mundo con aplicaciones ilimitadas. Fotografía extraída de la película *Ready Player One* (2018), basada en la novela del mismo nombre de Ernest Cline.

Una tendencia diferente, de empresas como Fotokite [1], proporciona a los UAVs un tiempo de vuelo casi ilimitado con una novedosa tecnología de anclaje, que combinada con el uso de imágenes térmicas permite asistir a los trabajadores en las tareas de bomberos y rescate. Además, muchas familias tienen acceso a robots de aspiradoras autónomas, como la Dyson 360 Eye [6], que puede ser fácilmente lanzada desde una aplicación en su teléfono, o a robots móviles de asistencia que apoyan a las personas mayores con estimulación cognitiva y social, asistencia y monitoreo transparente [101].

A pesar de las evidentes diferencias entre muchas de las aplicaciones mencionadas, todas ellas tienen en común la necesidad de conocer el entorno de aplicación y, lo que es más importante, todas ellas necesitan conocer con precisión su localización relativa en dicho escenario. Estos dos problemas, que en principio parecen desacoplados, han sido tradicionalmente abordados de forma simultánea en un conjunto de técnicas conocidas como *Localización y Mapeo Simultáneo* (de sus siglas en inglés *SLAM*) que ha sido formulado y resuelto en innumerables formas para diferentes aplicaciones.

## Motivación

En los últimos años, el SLAM ha jugado un papel de capital importancia en los rápidos avances tecnológicos en VR/AR/MR (AR y MR son las siglas de realidad *augmentada* y *mixta* respectivamente) y robótica, como parte vital de sus algoritmos de procesamiento, además de como base para el desarrollo en paralelo de técnicas más avanzadas como la evitación de obstáculos, el

reconocimiento de objetos, la planificación de tareas, el mapeo semántico, y un largo etcétera. Como su nombre indica, el SLAM comprende la estimación del *estado* de un robot mientras, simultáneamente, se realiza la construcción incremental y refinamiento de una representación consistente del entorno, o como se le conoce, el llamado *mapa*, basado en los sensores equipados. El estado del robot se describe normalmente por su *pose*, que está formada por la *posición y orientación* ya sea en 2D o 3D, aunque también puede contener diferentes aspectos en función de la aplicación, como la *velocidad y aceleración* del robot o parámetros de los sensores como las *calibraciones* o los *errores sistemáticos* de los mismos.

Por otra parte, el mapa abarca algunos aspectos de interés que representan el entorno operado por el robot y, por lo tanto, depende en gran medida de los sensores seleccionados. En consecuencia, existe una amplia variedad de representaciones observables del entorno, tanto en lo que se refiere a la aplicación como a la selección del sensor. Por ejemplo, un *mapa de ocupación* puede describir el entorno en una aplicación de vigilancia o para un robot de limpieza equipado con un *sensor láser*, mientras que un mapa de *características tridimensionales* extraídas de una cámara montada en un dron puede utilizarse en tareas de rescate o asistencia en incendios. Al mismo tiempo, la información del mapa se emplea en la estimación del estado del robot, reduciendo así la deriva de la localización a lo largo del tiempo que los enfoques más sencillos, como la *odometría* o la *localización con baliza*, cometen rápidamente gracias a la ventaja de volver a visitar y reconocer áreas previamente mapeadas, en lo que se conoce como *cierre de bucle*.

Hoy en día, las técnicas de *SLAM visual*, es decir, las que emplean algún tipo de cámara, han alcanzado una gran madurez con resultados impresionantes en entornos controlados. De hecho, desde un punto de vista teórico y conceptual, la comunidad científica ha considerado el SLAM como un problema resuelto desde la última década y, sin embargo, hoy en día es uno de los temas de investigación más activos en *visión por computador y robótica* y su popularidad no para de crecer. Por supuesto, una de las razones es el enorme abismo entre la perspectiva teórica y el problema real con los datos procedentes de los sensores reales y los inconvenientes del mundo real no previstos de antemano en los algoritmos de control de los sistemas. Considerado esto, queda un largo antes de lograr una solución SLAM robusta para situaciones reales, tales como entornos *dinámicos, poco texturados* o con *iluminaciones dinámicas o complejas, cambios en apariencia* en aplicaciones persistentes, *escalabilidad* de los datos procesados, o incluso un entendimiento de más alto nivel, por ejemplo *semántico*, del entorno.

## Contribuciones

Esta Tesis Doctoral contribuye a superar algunas de las limitaciones antes mencionadas de las técnicas tradicionales de SLAM visual y/o odometría, abor-

dando el problema desde diferentes perspectivas. Concretamente, esta tesis pretende avanzar a través de un sistema SLAM visual robusto que mitigue la limitación de las técnicas actuales, es decir, la robustez a diferentes tipos de entornos, iluminaciones complejas, etc. En este contexto, el alcance de esta Tesis Doctoral comprende por un lado el diseño e implementación de nuevos algoritmos de percepción y navegación que proporcionen una localización precisa y algún tipo de representación del entorno y, por otro, la integración de estos enfoques junto con tecnologías en aplicaciones del mundo real, como la robótica móvil.

De esta manera, los principales aportes de esta Tesis Doctoral se pueden agrupar en dos grandes temas que se describen a continuación:

### **Contribuciones al SLAM en entornos pocos texturados.**

El primer conjunto de trabajos, presentados en [61, 62, 65, 123], se centra en mejorar la robustez de la odometría visual y las técnicas SLAM en entornos con poca textura donde es habitual que el rendimiento de los enfoques tradicionales disminuya debido a las dificultades para encontrar un número suficiente de puntos fiables. El efecto, en estos casos, es un empobrecimiento de la precisión y, ocasionalmente, el fallo total del sistema. Por el contrario, muchos de estos entornos de baja textura contienen elementos planares que son ricos en formas lineales, por lo que una opción de característica alternativa como segmentos explotaría la información de partes estructuradas de la escena.

En este contexto, primero se ha contribuido con [62] un sistema completo de odometría estéreo probabilística que, gracias a la combinación de puntos y segmentos, fuera capaz de trabajar de forma robusta en entornos tan difíciles. Desafortunadamente, el tratamiento de los segmentos en las imágenes no es tan sencillo como en el caso de los puntos característicos, ya que son difíciles de representar a la vez que requieren una mayor carga computacional para su detección y seguimiento, lo que aumenta la complejidad del problema. Además, la contribución presentada en [123] empleó este sistema de odometría estéreo para desarrollar y probar un *modelo robusto probabilístico* para los errores de proyección de características puntuales basadas en datos reales mediante su modelado con *distribuciones Gamma* que mejoró tanto la precisión como la exactitud del sistema.

Para rebajar estas dificultades adicionales, en [61] se extendió un popular enfoque *semi-directo* a la odometría visual monocular, conocida como SVO [53], para explotar también la información de los segmentos, obteniendo así un sistema más robusto capaz de tratar tanto con entornos texturizados como con entornos estructurados. Como consecuencia directa, el sistema propuesto permitía un seguimiento más rápido de las características, ya que dicho enfoque eliminaba la necesidad de extraer y emparejar continuamente las características entre las diferentes imágenes de las secuencias.

Finalmente, esta Tesis Doctoral también contribuye con PL-SLAM [65], un sistema de SLAM visual para cámaras estéreo en tiempo real, que combina tanto puntos como segmentos para trabajar de forma robusta en una mayor variedad de escenarios. En esta contribución, la importancia de ambos tipos de características se aprovecha en todas las instancias del proceso: odometría visual, selección de *keyframes*, ajuste de haces, etc. Además, contribuye con un procedimiento de cierre de bucle a través de un novedoso enfoque de "bolsa de palabras" que explota el poder descriptivo combinado de los dos tipos de características. Además, el mapa resultante es más rico y diverso en elementos 3D, que pueden ser explotados para inferir valiosas estructuras de escena de alto nivel como planos, espacios vacíos, el plano del suelo, etc.

Los algoritmos desarrollados se han comparado con otras soluciones del estado del arte en conjuntos de datos y referencias conocidas y disponibles públicamente. Además, se publicó una implementación en C++ de código abierto de los algoritmos propuestos junto con los artículos publicados y material multimedia adicional para el beneficio de la comunidad científica.

## **Contribuciones al SLAM en entornos con iluminaciones dinámicas o complejas.**

En este segundo grupo de contribuciones, presentadas en [63, 66], se abordó uno de los principales retos abiertos en odometría visual y SLAM, es decir, su robustez frente a condiciones de iluminación difíciles o entornos de alto rango dinámico (HDR). Las principales dificultades en estas situaciones provienen tanto de las limitaciones de los sensores como de la incapacidad de realizar un seguimiento exitoso de los puntos de interés debido a las suposiciones audaces en SLAM, tales como constancia de luminosidad. El trabajo de esta Tesis Doctoral contribuye a este fenómeno desde dos perspectivas diferentes.

La primera contribución, presentada en [66], aborda este problema desde una perspectiva de aprendizaje profundo mediante la mejora de las imágenes monoculares, convirtiéndolas a representaciones más informativas e invariantes para VO y SLAM, aprovechando así las propiedades de generalización de las redes neuronales para lograr un rendimiento robusto en condiciones variadas. Este trabajo también demuestra cómo la inserción de capas de memoria de largo plazo (LSTM) nos permite obtener secuencias temporalmente consistentes, ya que la estimación depende de los estados previos. Las afirmaciones se validaron comparando el rendimiento de dos algoritmos del estado del arte en odometría/SLAM monocular (ORB-SLAM [115] y DSO [49]) comparando su rendimiento con la secuencia original y la mejorada, mostrando los beneficios de este enfoque en entornos difíciles.

En segundo lugar, una perspectiva más tradicional fue explotada en [63] donde un enfoque puramente *geométrico* para el *seguimiento robusto* de segmentos en secuencias estéreo complejas con cambios de iluminación severos o entornos de Alto Rango Dinámico (HDR). Esta contribución demuestra que,

gracias al hecho de que son más informativos, los segmentos pueden ser localizados con éxito a lo largo de las secuencias de vídeo considerando únicamente su consistencia geométrica a lo largo de imágenes continuas, y lo validaba evaluando tanto el rendimiento como la estimación de movimiento en secuencias de vídeo complejas a partir de conjuntos de datos de referencia en el estado del arte.

## Publicaciones

La presente tesis recoge las siguientes publicaciones, junto a material multimedia en muchos casos, en:

### Revistas

- *Ruben Gomez-Ojeda, Francisco-Angel Moreno, David Zuñiga-Noël, Davide Scaramuzza y Javier Gonzalez-Jimenez. **PL-SLAM: a Stereo SLAM System through the Combination of Points and Line Segments.** IEEE Transactions on Robotics (2019), Volumen 35(3), páginas 734-746. DOI: 10.1109/TRO.2019.2899783.  
Vídeo: [https://youtu.be/-lCTf\\_tAxxQ](https://youtu.be/-lCTf_tAxxQ)  
Código fuente: <https://github.com/rubengooj/pl-slam>*

### Conferencias Internacionales

- *Ruben Gomez-Ojeda y Javier Gonzalez-Jimenez. **Robust stereo visual odometry through a probabilistic combination of points and line segments.** En IEEE International Conference on Robotics and Automation (ICRA), Estocolmo, Suecia (2016), 2521-2526. DOI: 10.1109/ICRA.2016.7487406.  
Vídeo: <https://youtu.be/RIw7RCAY1II>  
Código fuente: <https://github.com/rubengooj/stvo-pl>*
- *Ruben Gomez-Ojeda, Jesus Briales y Javier Gonzalez-Jimenez. **PL-SVO: Semi-Direct Monocular Visual Odometry by Combining Points and Line Segments.** En IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Daejeon, Korea (2016), 4211-4216. DOI: 10.1109/IROS.2016.7759620.  
Vídeo: <https://youtu.be/c9hcKdSjtps>  
Código fuente: <https://github.com/rubengooj/pl-svo>*
- *Ruben Gomez-Ojeda, Francisco-Angel Moreno y Javier Gonzalez-Jimenez. **Accurate stereo visual odometry with gamma distributions.** En IEEE International Conference on Robotics and Automation (ICRA), Singapur (2017), 1423-1428. DOI: 10.1109/ICRA.2017.7989170.*

- *Ruben Gomez-Ojeda, Zichao Zhang, Javier Gonzalez-Jimenez y Davide Scaramuzza. Learning-based image enhancement for visual odometry in challenging HDR environments.* En IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia (2018), 805-811.  
DOI: 10.1109/ICRA.2018.8462876.  
*Vídeo:* [https://youtu.be/NKx\\_zi975Fs](https://youtu.be/NKx_zi975Fs)
- *Ruben Gomez-Ojeda y Javier Gonzalez-Jimenez. Geometric-based Line Segment Tracking for HDR Stereo Sequences.* En IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Madrid, España (2018), 69-74.  
DOI: 10.1109/IROS.2018.8593646.  
*Vídeo:* <https://youtu.be/VpdpS1tuRvc>

## Marco de la tesis

Esta tesis es el resultado de 5 años de trabajo del autor como miembro del grupo de investigación de Machine Perception and Intelligent Robotics (MAPIR) del Departamento de Ingeniería de Sistemas y Automática de la Universidad de Málaga. Esta investigación ha sido financiada principalmente por el programa de becas FPI (Formación de Personal Investigador), apoyado por el Ministerio de Economía y Competitividad de España.

Durante este período, el autor completó el programa de doctorado en Ingeniería Mecatrónica en el Departamento de Ingeniería de Sistemas y Automática, donde obtuvo un sólido conocimiento en cuatro de los pilares fundamentales de la robótica: sistemas de control, sistemas electrónicos, sistemas mecánicos y computadores. El programa de doctorado también se completó con varios cursos técnicos, como "Escritura Científica" en la Universidad de Málaga, y con la participación en la "Escuela Internacional de Verano en Visión por Computador", celebrada en Sicilia en 2016, que tenía como objetivo proporcionar una gran oportunidad para interactuar a jóvenes investigadores y estudiantes de doctorado, además de una interacción directa y debates con líderes mundiales en el campo de la Visión por Computador.

Desde Octubre de 2016 hasta Febrero de 2017 el autor realizó una estancia de investigación en el Grupo Robotics and Perception Group (RPG)<sup>1</sup> perteneciente tanto a la Universidad de Zurich como a la ETH Zurich bajo la supervisión del Prof. Dr. Davide Scaramuzza. Más recientemente, el autor fue becario de investigación en Zurich en Oculus VR (Facebook) durante el verano de 2018, a la que posteriormente se incorporó de nuevo en febrero de 2019, bajo la supervisión directa de Matia Pizzoli, donde también tuvo la oportunidad de colaborar en proyectos de categoría mundial en realidad virtual colaborando con Christian Forster, Michael Burri y Luc Oth, entre otros.

<sup>1</sup><http://rpg.ifi.uzh.ch/>

Además, el autor ha sido revisor de artículos de prestigiosas conferencias y revistas, tales como las conferencias IEEE International Conference on Robotics and Automation (ICRA, 2016, 2017, 2018, 2019), IEEE International Conference on Intelligent Robots and Systems (IROS, 2015, 2017, 2018), o las revistas IEEE Robotics and Automation Letters (RA-L, 2017, 2018, 2019), International Journal of Robotics Research (IJRR, 2018), o IEEE Transactions on Robotics (T-RO, 2019).

La beca FPI también ha ofrecido la oportunidad de colaborar como asistente de laboratorio con el Departamento de Ingeniería de Sistemas y Automática. Durante el trabajo de esta Tesis Doctoral, el autor impartió durante dos años las asignaturas de "Programación Robótica" en la Facultad de Informática de la Universidad de Málaga, y "Diseño de Controladores Industriales" en la Escuela de Ingeniería Industrial de la Universidad de Málaga. El autor también fue co-supervisor de la Tesis de Máster de David Zúñiga Noël, titulada "SLAM based on Depth Sensors".

Además de la investigación en el ámbito de esta Tesis Doctoral, el autor ha participado en otros proyectos del grupo MAPIR, algunos de ellos con temas relacionados:

- **Taroth: New developments toward a Robot at Home:** en este proyecto se abordan los tres objetivos siguientes: 1) mejorar la fiabilidad del movimiento del robot, 2) integrar y explotar la semántica para mejorar la autonomía del robot y su interacción con los seres humanos, y 3) desarrollar una arquitectura de software robótico que pueda gestionar los servicios de *Ambient Assisted Living* relacionados con el entretenimiento, la domótica, las redes sociales, la seguridad, etc.
- **GiraffPlus: Combining social interaction and long term monitoring for promoting independent living (FP7-ICT-2011-7).** Este proyecto propone la creación de un entorno inteligente para registrar de manera continua (24/7) datos de la actividad de la persona en la casa así como de sus parámetros fisiológicos, ofreciendo a partir de ellos información relevante y personalizada para su médico, enfermero y cuidadores. El sistema se completa con un robot en la vivienda que permite la visita de éstos.
- **PROMOVE: Avances en robótica móvil para promover la vida independiente de personas mayores (DPI2014-55826-R),** cuyo objetivo es avanzar hacia un robot personal que facilite y prolongue la vida independiente de personas mayores en sus domicilios. Para ello, se pretende paliar las deficiencias y limitaciones que aún existen en el estado del arte de la robótica móvil y al percepción artificial, proponiendo soluciones más robustas, eficientes y efectivas.

Así mismo, de la colaboración del autor de esta tesis en éstos proyectos han surgido las siguientes publicaciones:

### Revistas

- *Manuel Lopez-Antequera, Ruben Gomez-Ojeda, Nicolai Petkov y Javier Gonzalez-Jimenez. Appearance-invariant place recognition by discriminatively training a Convolutional Neural Network.* Pattern Recognition Letters (2017) 92, 89-95.  
DOI: 10.1016/J.PATREC.2017.04.017.
- *David Zúñiga-Noel, Jose-Raul Ruiz-Sarmiento, Ruben Gomez-Ojeda, and Javier Gonzalez-Jimenez. Automatic Multi-Sensor Extrinsic Calibration for Mobile Robots.* IEEE Robotics and Automation Letters (2019), Volumen 4 (3), 2862 - 2869.  
DOI: 10.1109/LRA.2019.2922618.  
Source code: [https://github.com/dzunigan/robot\\_autocalibration](https://github.com/dzunigan/robot_autocalibration)
- *David Zúñiga-Noël, Alberto Jaenal, Ruben Gomez-Ojeda, and Javier Gonzalez-Jimenez. The UMA-VI Dataset: Visual-Inertial Odometry in Low-textured and Dynamic Illumination Environments.* Aceptado en International Journal of Robotics Research (2020).  
Dataset: <http://mapir.uma.es/work/uma-visual-inertial-dataset>.

### Conferencias

- *Ruben Gomez-Ojeda, Jesus Briales, Eduardo Fernandez-Moral y Javier Gonzalez-Jimenez. Extrinsic calibration of a 2d laser-rangefinder and a camera based on scene corners.* En IEEE International Conference on Robotics and Automation (ICRA), Seattle, Estados Unidos (2015), 3611-3616.  
DOI: 10.1109/ICRA.2015.7139700.  
Video: <https://youtu.be/frRKTZ1utJ0>
- *David Zúñiga-Noel, Ruben Gomez-Ojeda, Francisco-Angel Moreno y Javier Gonzalez-Jimenez. Calibración extrínseca de un conjunto de cámaras RGB-D sobre un robot móvil.* En XXXVIII Jornadas de Automática (2017).

### Talleres Internacionales

- *Ruben Gomez-Ojeda. Visual Odometry and SLAM using Line Segment Features.* Ponente invitado en International Workshop on Lines, Planes and Manhattan Models for 3-D Mapping (LPM17), como parte del 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) en Vancouver, Canadá (2017).

## Informes Técnicos

- *Ruben Gomez-Ojeda, Manuel Lopez-Antequera, Nicolai Petkov y Javier Gonzalez-Jimenez. Training a convolutional neural network for appearance-invariant place recognition.* Universidad de Málaga (2015). arXiv preprint arXiv:1505.07428.

## Estructura de la Tesis

Además de este resumen en Castellano, el resto de la tesis está estructurada en los siguientes capítulos:

**Chapter 1: Introduction** resume los principales aspectos y contribuciones de la presente Tesis Doctoral.

**Chapter 2: Simultaneous Localization and Mapping** revisa el estado del arte de este importante problema y describe brevemente las técnicas más extendidas basadas en *características*, *métodos directos* y *métodos semi-directos*, situando este trabajo en el contexto del SLAM visual.

**Chapter 3: Robustness to Low-textured Environments** describe las dificultades que sufren la mayoría de las soluciones tradicionales en entornos poco texturados, en los que su precisión suele deteriorarse. Este capítulo también presenta tres contribuciones de la presente Tesis Doctoral, las dos primeras son enfoques puramente odométricos, en el contexto de SLAM en entornos con poca textura, todas ellas presentadas a lo largo de bibliotecas C++ disponibles públicamente.

**Chapter 4: Dealing with Dynamic Illumination and HDR Environments** trata uno de los principales desafíos abiertos en el SLAM visual, cambios bruscos de iluminación, donde las dificultades provienen tanto de las limitaciones de los sensores como de las suposiciones audaces que a menudo se introducen en los algoritmos para aliviar los requisitos computacionales. En este Capítulo se presentan dos contribuciones diferentes en este tema, una desde una perspectiva de *deep-learning*, y otra desde un punto de vista geométrico para permitir el seguimiento de las características en tales condiciones.

**Chapter 5: Dataset for Low-textured, Dynamic Illumination and HDR Environments** propone un dataset visual e inercial que contiene situaciones desafiantes, centrándose en entornos con poca textura

o iluminaciones difíciles y dinámicas. El objetivo de este conjunto de datos es proporcionar un punto de referencia para la evaluación de los algoritmos de odometría visual-inercial en estas situaciones, permitiendo evaluar la deriva acumulada estimada de la trayectoria con el fin de medir y comparar fácilmente los resultados de diferentes algoritmos.

**Chapter 6: Conclusions** ofrece algunas ideas finales extraídas del trabajo realizado en esta Tesis Doctoral y presenta brevemente las futuras líneas de investigación aún abiertas en relación con las contribuciones de este trabajo.

## Conclusiones y líneas futuras

En los últimos 20 años, las técnicas de SLAM visual han alcanzado una alta madurez con resultados impresionantes en entornos y ambientes controlados. De hecho, el SLAM ha sido considerado un problema teóricamente resuelto durante la última década, tal y como afirmaron Durrant-Whyte y Bailey en 2006 [43] (traducido del inglés): *A nivel teórico y conceptual, el SLAM puede considerarse un problema resuelto. Sin embargo, sigue habiendo problemas sustanciales en la implementación práctica de soluciones SLAM más generales y, en particular, en la construcción y el uso de mapas ricos en percepciones como parte de un algoritmo SLAM.*

Hoy, más de una década después, sigue siendo uno de los temas de investigación más activos en visión por computador y robótica móvil, y la cuestión de *está el problema de SLAM resuelto?* se plantea a menudo en la comunidad científica [24]. Una de las razones detrás de ello es que, a pesar de la gran madurez alcanzada por las técnicas visuales SLAM de última generación, todavía quedan muchos retos por resolver antes de llegar a un sistema SLAM robusto para operaciones a largo plazo en escenarios no controlados, donde las suposiciones clásicas, como los entornos estáticos, no se mantienen.

Esta tesis contribuye a superar algunas de las limitaciones antes mencionadas de las técnicas tradicionales de SLAM visual y/o odometría, abordando el problema desde diferentes perspectivas. Específicamente, este trabajo tiene como objetivo avanzar hacia un sistema visual SLAM robusto que mitigue la limitación de las técnicas actuales, es decir, la robustez a diferentes tipos de entornos, las iluminaciones desafiantes, etc. En este contexto, el alcance de esta Tesis Doctoral comprende, por un lado, el diseño e implementación de nuevos algoritmos de percepción y navegación que proporcionen una localización precisa y algún tipo de representación del entorno y, por otro, la integración de estos enfoques junto con tecnologías en aplicaciones del mundo real, como la robótica móvil. Las conclusiones principales de esta Tesis Doctoral se pueden agrupar en dos grandes temas que se describen a continuación:

*SLAM en entornos pocos texturados.* La primera serie de trabajos se ha centrado en la mejora de la robustez de la odometría visual y las técnicas SLAM en entornos de baja textura, donde es habitual que el rendimiento de los enfoques tradicionales disminuya debido a las dificultades para encontrar un número suficiente de características de punto fiables. En tales casos, el efecto es un empobrecimiento en la precisión y, ocasionalmente, el fallo total del sistema. Este grupo de trabajos se beneficia de una elección de características alternativas, es decir, líneas o segmentos, para explotar la información de las partes más estructuradas del entorno.

Para ello, esta Tesis Doctoral contribuye con un sistema SLAM en tiempo real para cámaras estéreo que aprovecha la importancia de ambos tipos de características que se aprovechan en todas las instancias del proceso: odometría visual, selección de *keyframes*, ajuste de haces, etc. Además, contribuye con un procedimiento de cierre de bucle mediante un novedoso enfoque de bolsa de palabras que explota el poder descriptivo combinado de los dos tipos de características. Entre los beneficios, el sistema resultante es más robusto en entornos difíciles y, además, los mapas estimados son más ricos y diversos en elementos 3D, que pueden ser explotados para inferir estructuras valiosas de alto nivel como planos, espacios vacíos, suelo, etc.

Por otro lado, el tratamiento de las características de los segmentos en las imágenes no es tan sencillo como en el caso de los puntos, ya que son difíciles de representar y, al mismo tiempo, requieren una mayor carga computacional para su seguimiento. Para aliviar estas dificultades adicionales, esta Tesis también se ha beneficiado del enfoque semi-directo de la odometría monocular para extender un trabajo previo del estado del arte, conocido como SVO [53], con el fin de trabajar simultáneamente con segmentos. Esto permitió un seguimiento más rápido de las características, ya que el planteamiento semi-directo eliminó la necesidad de una continua detección y seguimiento de las características.

Los algoritmos desarrollados han sido comparados con otras soluciones de vanguardia en conjuntos de datos extendidos y conocidos por la comunidad, y adicionalmente se ha publicado una implementación C++ de código abierto de los algoritmos propuestos, junto con los artículos publicados además de material multimedia adicional para el beneficio de la comunidad.

*SLAM en entornos con iluminaciones dinámicas o complejas.*

Uno de los principales retos abiertos en la odometría visual y SLAM es su *robustez* frente a condiciones de iluminación difíciles o entornos HDR. En tales casos, las dificultades provienen tanto de las limitaciones de los sensores, como de los cambios rápidos de áreas oscuras a áreas brillantes que pueden sobreexponer las imágenes, y de la incapacidad de realizar un seguimiento exitoso de los puntos de interés debido a las tradicionales hipótesis realizadas en SLAM, como la constancia local del brillo en las imágenes procesadas. El trabajo

de esta Tesis Doctoral contribuye a este fenómeno desde dos perspectivas diferentes.

En la primera, se ha abordado este problema desde una perspectiva de aprendizaje profundo (*deep-learning*) mediante la mejora de las imágenes monoculares a representaciones más informativas e invariantes para odometría y SLAM, ya que las redes neurales han demostrado alcanzar un rendimiento robusto en dichas condiciones gracias a sus propiedades de generalización. Este trabajo también ha demostrado cómo la inserción de redes de memoria a corto plazo (LSTM) permitía la obtención de secuencias temporalmente consistentes, ya que la estimación también dependía de estados previos.

En cambio, el segundo trabajo adoptó una perspectiva más tradicional desde un punto de vista puramente geométrica para el seguimiento de segmentos en secuencias de imágenes en escenarios con cambios de iluminación severos o entornos HDR. En esta contribución se demostró que, gracias a la naturaleza más informativa de las líneas con respecto a los puntos, éstas pueden ser detectadas y correspondidas a lo largo de secuencias de vídeo rápidamente, simplemente teniendo en cuenta restricciones geométricas.

Adicionalmente, con el fin de suplir la falta de datos específicos para los desafíos abiertos antes mencionados, esta Tesis Doctoral también aporta un conjunto de datos visuales-inerciales que contienen situaciones desafiantes del mundo real, especialmente entornos con poca textura e iluminación dinámica. Con este conjunto de datos, pretendemos suplir esta falta de datos, permitiendo así evaluaciones y comparaciones exhaustivas de los métodos en tales condiciones, y ayudar al desarrollo de técnicas más robustas en entornos no controlados.

## Trabajos futuros

Aparte de las mejoras en la robustez, que motivaron este trabajo de tesis, hay muchos otros temas interesantes para lograr un sistema SLAM fiable capaz de trabajar en escenarios arbitrarios:

*SLAM basada en apariencia en largo plazo.* Un enfoque diferente del problema del SLAM visual con respecto a los resumidos en Chapter 2, que en definitiva estima la pose relativa con respecto a un mapa, se basa en apariencia. Uno de los beneficios de este enfoque es su robustez ante cambios visuales drásticos, tales como las que se producen entre secuencias que se toman durante el día y la noche, los cambios estacionales, o cambios estructurales a largo plazo. Por el contrario, en la actualidad estos métodos no son lo suficientemente fiables para la relocalización métrica, donde los enfoques basados en características siguen siendo los mejores, pero sin embargo no proporcionan invarianza a tales cambios dramáticos en la apariencia.

*SLAM Activo.* Una línea de investigación emergente que, en general, trata de emplear la información de entrada para predecir automáticamente los ajustes óptimos para cada situación. Por ejemplo, la mayoría de las implementaciones de SLAM requieren un ajuste extensivo de parámetros, lo cual típicamente se hace de manera empírica para un escenario dado, lo cual puede no ser suficiente en escenarios arbitrarios. Otros ejemplos pueden ser el control de los parámetros de la cámara con valores predichos para maximizar el rendimiento de la misma, o proporcionar a los robots con cámaras móviles capaces de predecir las partes del mapa más informativas para la tarea asignada.

*Mapas Semánticos.* Los métodos SLAM típicos consisten en un conjunto de puntos de referencia 3D que pueden ser utilizados para tareas de robótica como la evitación de obstáculos o la navegación, pero su principal ventaja es la reducción de los errores modelados para una localización más precisa. Por otro lado, este tipo de mapas están muy limitados para realizar, por ejemplo, tareas más complejas como el reconocimiento de objetos/personas, o misiones robóticas de nivel superior, como "Ir a la cocina", donde se requiere un conocimiento contextual del entorno. Para ilustrar esto, una aplicación autónoma de SLAM para coches podría beneficiarse del uso de información semántica del entorno para predecir la pose del robot a partir de los objetos estáticos mientras que al mismo tiempo estimando el estado de los objetos dinámicos.

*Deep-learning in SLAM.* Finalmente, otro grupo de técnicas está empezando a emerger en la comunidad SLAM, es decir, aquellos que emplean *deep-learning* para proporcionar a los sistemas un conocimiento de alto nivel, que difícilmente se puede lograr con técnicas puramente geométricas [36]. Por ejemplo, en [105] los autores combinan las CNN con SLAM geométrico, para proporcionar mapas 3D semánticamente etiquetados en tiempo real. El *deep-learning* también se ha utilizado para mejorar las técnicas tradicionales de SLAM, *e.g.* [21], o incluso para proponer un sistema de SLAM denso con estimación de mapas de profundidad enteramente aprendido [156].

## Introduction

*Imagine* for a moment that people could wear *virtual reality (VR)* devices that allowed them to join a world with limitless possibilities, where for instance an engineer could have visual feedback of the model being designed, surgeons could rehearse their following surgeries with a realistic *3D model* of the patient, or a person could simply walk through a city and witness the recreation of some of its historic moments. Furthermore, imagine that an *autonomous car* could drive you to work and gather up your groceries moments before commuting you back home, or *unmanned autonomous vehicles (UAVs)* played a vital role in disaster assistance, or smart and *telepresence robots* could aid elderly people in their daily tasks.

Not so long ago this conceptions were considered as unrealistic and futuristic ideas only worth of science fiction tales as in Figure 1.1 , but as a matter of fact, most of them are a reality nowadays, or at least it is not bold to assume they will become one over the following years. In fact, there exist a few VR companies, such as Oculus VR [5], that allow people to immerse themselves to 3D movies, video-games or Google Earth [2] VR walks while interacting with the virtual environment. There are also a number of companies, such as Waymo [8] (formerly known as the Google self-driving car project [3]), Tesla [7] or Nuro [4], presenting their different implementations of *self-driving cars* that are currently being tested in cities like San Francisco. A different trend, from companies such as Fotokite [1], provides almost unlimited flight time to UAVs with a novel tethering technology combined with thermal images with the aim of aiding workers in fire and rescue tasks. Moreover, many families have access to *autonomous vacuum cleaner robots*, such as the Dyson 360 Eye [6], that can be easily launched from an application in your smart-phone, or to *assistive mobile robots* that support elderly people with cognitive and social stimulation, assistance, and transparent monitoring [101].



**Figure 1.1:** Nowadays it is not daring to imagine a future where one can wear some virtual reality equipment and transport oneself to a world with limitless applications. Photograph extracted from the movie *Ready Player One* (2018) based on Ernest Cline's novel of the same name.

In spite of the evident differences between many of the aforementioned applications, they all require the knowledge and some *map representation* of the surrounding environment and, more importantly, they all need to accurately know their relative *localization* (position and orientation) in such scenario. These two, in principle, separated problems have been traditionally addressed jointly in a set of techniques known as *Simultaneous Localization and Mapping* (SLAM) that has been formulated and solved for uncountable sensor configurations and in a large number of manners.

## 1.A Motivation

In the last years, SLAM has played a role of capital importance in the rapid technological advances in VR/AR/MR (AR and MR stands for *augmented* and *mixed reality* respectively) and robotics, as a vital part of their processing pipelines and as baseline to the parallel development of more advanced techniques such as obstacle avoidance, object recognition, task planning, semantic mapping, and a long etcetera. As its name indicates, it comprises the estimation of the *state* of a robot and, simultaneously, the incremental construction and refinement of a consistent representation of the environment, *i.e.* the so-called *map*, based on the equipped sensors. The robot state is usually described by its *pose*, formed by the 2D/3D *position and orientation*, although different aspects can be considered regarding the specific application, such as *velocity* and *acceleration*, or sensor parameters, for instance the *biases* or the *intrinsic and extrinsic calibration* between the on-board sensors.

On the other hand, the map encompasses some aspects of interest representing the environment operated by the robot, and hence, it heavily depends on the selected sensors. In consequence, there exist a wide variety of representations of the environment regarding both the application and sensor selection. For instance, an *occupancy-grid map* can describe the scene for a surveillance or a cleaning robot equipped with a *laser range-finder*, while a map comprised by *3D features* (for instance points) extracted from a *camera* attached to a drone could be used in fire and rescue tasks. At the same time, the map information is further employed in the robot state estimation, thus reducing the drift over time that more simple approaches, such as odometry or dead-reckoning, quickly commit thanks to the benefit of re-visiting map areas, in what is known as *loop closure*.

Nowadays, *visual SLAM* techniques, *i.e.*, those employing some sort of camera, have reached a maturity with impressive results achieved in controlled environments. In fact, from a theoretical and conceptual level, the scientific community has even considered SLAM as a solved problem since the past decade [43], and yet, nowadays it is one of the most active research topics in *computer vision* and *robotics* and its popularity keeps growing. Of course, one of the reasons behind it is the huge abyss between the theoretical perspective and the real problem with data coming from actual sensors and real world unplanned inconveniences. With this considered, there are remaining issues before achieving a robust SLAM solution for real situations, such as highly *dynamic environments*, *low-textured* or *feature-deprived* scenarios, *challenging illuminations*, *long-term* or *appearance changes*, *life-long maintenance* and *scalability* of the processed data, or even higher level understanding of the maps from a *semantic* perspective.

## 1.B Contributions

This thesis contributes to overcome some of the aforementioned limitations of traditional visual SLAM and/or odometry techniques by addressing the problem from different perspectives. Concretely, this thesis aims to advance the state of the art through a *robust* visual SLAM system that mitigates the limitation of current techniques, *i.e.*, robustness to different types of environment, challenging illuminations, etc. In this context, the scope of the thesis comprehends on one hand the design and implementation of new perception and navigation algorithms that provide accurate localization and some type of representation of the environment, and, on the other, the integration of such approaches along with technologies in real world applications, such as mobile robotics.

Thereby, the main contributions of this thesis can be grouped into two major topics described in the following:

### 1.B.1 Contributions to SLAM in Low-textured Environments

The first set of works, presented in the papers [61,62,65,123] focuses on improving the robustness of visual odometry and SLAM techniques in low-textured environments, where it is common that the performance of traditional approaches decreases due to difficulties in finding a sufficient number of reliable point features. The effect in such cases is an accuracy impoverishment and, occasionally, the complete failure of the system. In contrast, many of such low-textured environments contain planar elements that are rich in straight shapes, so an alternative feature choice such as *line segments* would exploit information from structured parts of the scene.

In this context, we first contribute in [62] with a complete *probabilistic stereo visual odometry* system that, thanks to the combination of both points and line segments, was capable of robustly working in such difficult environments. Unfortunately, dealing with line segment features in images is not as straightforward as the case of point features, since they are difficult to represent while also requiring *higher computational burden* for its detection and tracking, hence increasing the complexity of the problem. Moreover, the contribution presented in [123] employed this stereo visual odometry system to develop and test a *robust probabilistic model* for the projection errors of point features based on real data by modeling them with *Gamma distributions* which improved both precision and accuracy of the system.

To alleviate these additional difficulties, we extended in [61] a popular *semi-direct* approach to monocular visual odometry, known as SVO [53], to also exploit information from line segments, hence obtaining a more robust system capable of dealing with both textured and structured environments. As a direct consequence, the proposed system allowed for faster feature tracking, since the semi-direct framework eliminated the necessity of continuously extracting and matching features between subsequent frames.

Finally, this thesis also contributes with PL-SLAM [65], a *real-time* stereo visual SLAM system that combined both points and line segments to work robustly in a wider variety of scenarios. In this contribution, the importance of both type of features are leveraged at all instances of the process: *visual odometry*, *keyframe selection*, *bundle adjustment*, etc. Moreover, it contributes with a *loop closure* procedure through a novel *bag-of-words* approach that exploits the combined descriptive power of the two kinds of features. Additionally, the resulting map is richer and more diverse in 3D elements, which can be exploited to infer valuable, high-level scene structures like planes, empty spaces, ground plane, etc.

The developed algorithms have been compared with other state-of-art solutions in well-known and publicly available datasets and benchmarks. Additionally, an *open-source C++ implementation* of the proposed algorithms was

released along with the published articles and some extra multimedia material for the benefit of the community.

### 1.B.2 Contributions to SLAM under Dynamic Illumination and HDR Environments

In this second group of contributions, presented in [63,66], we deal with one of the main open challenges in visual odometry and SLAM, *i.e.* its *robustness* to difficult *illumination* conditions or *high dynamic range (HDR)* environments. The main difficulties in these situations come from both the limitations of the sensors and the inability to perform a successful tracking of interest points because of some bold assumptions in SLAM, such as *brightness constancy*. The work of this thesis contributes to mitigate these phenomena from two different perspectives.

The first contribution, presented in [66], addresses this problem from a *deep learning* perspective by enhancing images to more informative and invariant representations for VO and SLAM, hence taking advantage of the *generalization* properties of deep neural networks to achieve robust performance in varied conditions. This work also demonstrates how the insertion of *long short term memory (LSTM)* allows us to obtain temporally consistent sequences, since the estimation depends on previous states. The claims are validated by comparing the performance of two state-of-art algorithms in monocular VO/SLAM (ORB-SLAM [115] and DSO [49]) with the original input and the enhanced sequences, showing the benefits of this approach in challenging environments.

Secondly, a more traditional perspective was exploited in [63], where a purely *geometric* approach for the *robust matching* of line segments for challenging stereo streams with severe illumination changes or High Dynamic Range (HDR) environments was proposed. This contribution claims that, thanks to the fact that line segments are more informative, they can be successfully tracked along video sequences by only considering their geometric consistency along consecutive frames. The proposed approach was validated by evaluating both the matching performance and motion estimation in challenging video sequences from benchmark datasets.

### 1.B.3 Publications

The present thesis encompasses the following publications and, in some of the cases, the *source code* and some demonstrative *videos* associated to them:

#### Journals

- *Ruben Gomez-Ojeda, Francisco-Angel Moreno, David Zuñiga-Noël, Davide Scaramuzza, and Javier Gonzalez-Jimenez. **PL-SLAM: a Stereo SLAM System through the Combination of Points and Line Segments.** IEEE Transactions on Robotics (2019), volume 35(3), pages 734-746. DOI: 10.1109/TRO.2019.2899783.  
Video: [https://youtu.be/-lCTf\\_tAxhQ](https://youtu.be/-lCTf_tAxhQ)  
Source code: <https://github.com/rubengooj/pl-slam>*

#### International Conferences

- *Ruben Gomez-Ojeda and Javier Gonzalez-Jimenez. **Robust stereo visual odometry through a probabilistic combination of points and line segments.** In IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden (2016), 2521-2526.  
DOI: 10.1109/ICRA.2016.7487406.  
Video: <https://youtu.be/RIw7RCAY1II>  
Source code: <https://github.com/rubengooj/stvo-pl>*
- *Ruben Gomez-Ojeda, Jesus Briales, and Javier Gonzalez-Jimenez. **PL-SVO: Semi-Direct Monocular Visual Odometry by Combining Points and Line Segments.** In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Daejeon, Korea (2016), 4211-4216.  
DOI: 10.1109/IROS.2016.7759620.  
Video: <https://youtu.be/c9hcKdSjtps>  
Source code: <https://github.com/rubengooj/pl-svo>*
- *Ruben Gomez-Ojeda, Francisco-Angel Moreno, and Javier Gonzalez-Jimenez. **Accurate stereo visual odometry with gamma distributions.** In IEEE International Conference on Robotics and Automation (ICRA), Singapore (2017), 1423-1428.  
DOI: 10.1109/ICRA.2017.7989170.*
- *Ruben Gomez-Ojeda, Zichao Zhang, Javier Gonzalez-Jimenez, and Davide Scaramuzza. **Learning-based image enhancement for visual odometry in challenging HDR environments.** In IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia (2018), 805-811.  
DOI: 10.1109/ICRA.2018.8462876.  
Video: [https://youtu.be/NKx\\_zi975Fs](https://youtu.be/NKx_zi975Fs)*

- *Ruben Gomez-Ojeda and Javier Gonzalez-Jimenez. Geometric-based Line Segment Tracking for HDR Stereo Sequences.* In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Madrid, Spain (2018), 69-74.  
DOI: 10.1109/IROS.2018.8593646.  
Video: <https://youtu.be/VpdpS1tuRvc>

## 1.C Framework and Timeline

This thesis is the result of five years of work by the author as a member of the *Machine Perception and Intelligent Robotics (MAPIR)* research group<sup>1</sup>, which is part of the Department of System Engineering and Automation of the University of Malaga, where the author started working under the supervision of Prof. Javier González Jiménez in the Master Thesis entitled *A probabilistic approach to stereo visual odometry based on line segments*. After this, the author received an FPI grant (*Formación de Personal Investigador*), supported by the Spanish Ministry of Economy and Competitiveness, which mainly funded this doctoral research.

During this period, the author completed the doctoral program in *Mechatronics Engineering* at the Department of System Engineering and Automation where he acquired a strong background knowledge concerning the four fundamental pillars of robotics: control systems, electronic systems, mechanical systems, and computers. The doctoral program was also completed with several technical courses, such as “Scientific Writing” at the University of Malaga, and with the participation in the “*International Computer Vision Summer School*”, held in Sicily in 2016, which aimed to provide a stimulating opportunity for young researchers and Ph.D. students with direct interaction and discussions with world leaders in the field of Computer Vision.

From October 2016 to February 2017 the author completed a research stay at the *Robotics and Perception Group (RPG)*<sup>2</sup> belonging to both the University of Zurich and ETH Zurich under the supervision of Prof. Dr. Davide Scaramuzza. More recently, the author was a research intern at *Oculus VR (Facebook)* in Zurich during the summer of 2018 under the direct supervision of Dr. Matia Pizzoli, and he joined again in February 2019. At Oculus he also had the opportunity of collaborating in world class projects in Virtual Reality with Christian Forster, Michael Burri and Luc Oth.

Additionally, the author has been an active reviewer of manuscripts from prestigious conferences and journals, such as the *IEEE International Conference on Robotics and Automation* (ICRA, 2016, 2017, 2018, 2019, 2020), the *IEEE International Conference on Intelligent Robots and Systems* (IROS, 2016, 2017, 2018, 2019), the *IEEE Robotics and Automation Letters* (RA-L,

<sup>1</sup><http://mapir.isa.uma.es/>

<sup>2</sup><http://rpg.ifi.uzh.ch/>

2017, 2018, 2019, 2020), the *International Journal of Robotics Research* (IJRR, 2018), or the *IEEE Transactions on Robotics* (T-RO, 2019).

The FPI grant also offered the opportunity to collaborate as a *laboratory assistant* with the Department of System Engineering and Automation. During this thesis work, the author taught for two years “Robotics Programming” at the Computer Science Faculty in the University of Malaga, and “Design of Industrial Controllers” at the Engineering School in the University of Malaga. The author was also co-supervisor of the Master’s Thesis of David Zúñiga Noël, entitled *SLAM based on Depth Sensors*.

In addition to the research in the scope of this thesis, the author has been also involved in other projects within the MAPIR group, some of them with related topics:

- **Taroth: New developments toward a Robot at Home:** in this project the three following targets are addressed: 1) improving dependability of the robot motion, 2) integrating and exploiting semantics to improve robot autonomy and interaction with humans, and 3) developing a robot software architecture that can manage Ambient Assisted Living services related to entertainment, domotics, social networking, safety, etc.
- **GiraffPlus: Combining social interaction and long term monitoring for promoting independent living:** this project pursues the creation of an intelligent environment to continuously record (24/7) data derived from the person’s activity at home and from their physiological parameters, and offering through them relevant and personalized information for his doctor, nurse, and/or person in charge. The system is completed with a telepresence robot in the house.
- **PROMOVE: Advances in mobile robotics to promote the independent life of elderly people:** the goal of this project is to advance through a personal robot that eases and prologues the independent life of elderly people. For that, this projects proposes more robust, efficient and effective solutions to mitigate the already existent limitations in the state of the art of mobile robotics.

From the author’s work in these projects arose a number of additional publications:

### Journals

- *Manuel Lopez-Antequera, Ruben Gomez-Ojeda, Nicolai Petkov, and Javier Gonzalez-Jimenez. Appearance-invariant place recognition by discriminatively training a Convolutional Neural Network. Pattern Recognition Letters* (2017) 92, 89-95.  
DOI: 10.1016/J.PATREC.2017.04.017.

- *David Zúñiga-Noel, Jose-Raul Ruiz-Sarmiento, Ruben Gomez-Ojeda, and Javier Gonzalez-Jimenez. Automatic Multi-Sensor Extrinsic Calibration for Mobile Robots.* In IEEE Robotics and Automation Letters (2019), Volume 4, Issue 3, 2862 - 2869.  
DOI: 10.1109/LRA.2019.2922618.  
Source code: [https://github.com/dzunigan/robot\\_autocalibration](https://github.com/dzunigan/robot_autocalibration)
- *David Zúñiga-Noel, Alberto Jaenal, Ruben Gomez-Ojeda, and Javier Gonzalez-Jimenez. The UMA-VI Dataset: Visual-Inertial Odometry in Low-textured and Dynamic Illumination Environments.* Accepted in International Journal of Robotics Research (2020).  
Dataset: <http://mapir.uma.es/work/uma-visual-inertial-dataset>.

### Conferences

- *Ruben Gomez-Ojeda, Jesus Briales, Eduardo Fernandez-Moral, and Javier Gonzalez-Jimenez. Extrinsic calibration of a 2d laser-rangefinder and a camera based on scene corners.* In IEEE International Conference on Robotics and Automation (ICRA), Seattle, United States (2015), 3611-3616.  
DOI: 10.1109/ICRA.2015.7139700.  
Video: <https://youtu.be/frRKTZ1utJ0>
- *David Zúñiga-Noel, Ruben Gomez-Ojeda, Francisco-Angel Moreno, and Javier Gonzalez-Jimenez. Calibración extrínseca de un conjunto de cámaras RGB-D sobre un robot móvil.* In XXXVIII Jornadas de Automática (2017).

### International Workshops

- *Ruben Gomez-Ojeda. Visual Odometry and SLAM using Line Segment Features.* Invited speaker at the International Workshop on Lines, Planes and Manhattan Models for 3-D Mapping (LPM17), as part of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) in Vancouver, Canada (2017).

### Technical reports

- *Ruben Gomez-Ojeda, Manuel Lopez-Antequera, Nicolai Petkov, and Javier Gonzalez-Jimenez. Training a convolutional neural network for appearance-invariant place recognition.* University of Malaga (2015).  
arXiv preprint arXiv:1505.07428.

## 1.D Outline

The rest of this thesis is organized as follows (Figure 1.2 includes a diagram with the structure as well):

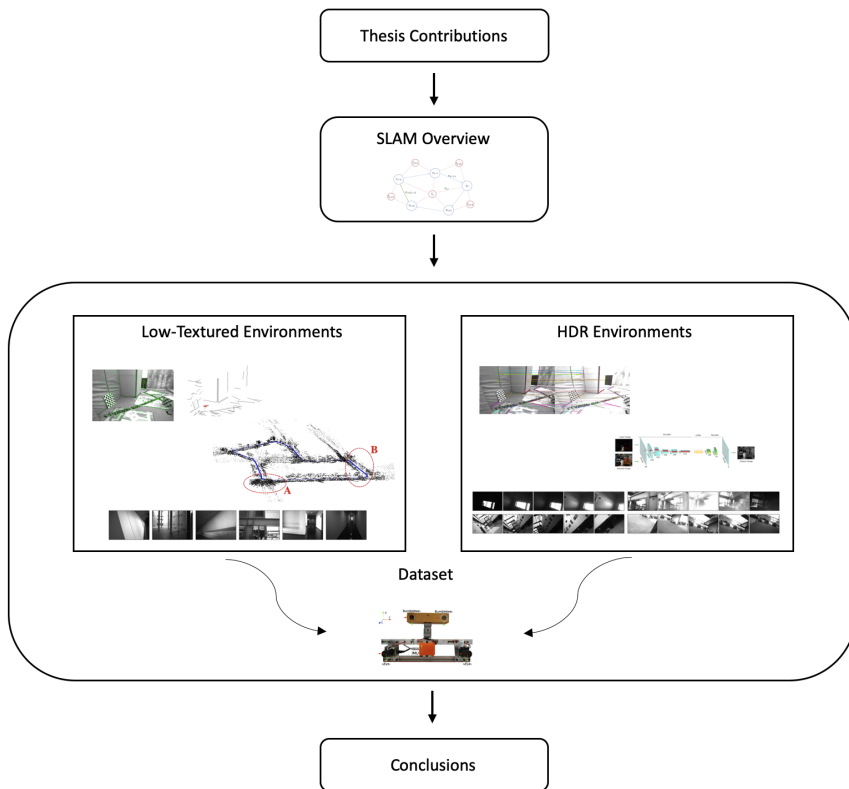
**Chapter 2: Simultaneous Localization and Mapping** reviews the state of the art of this important problem and briefly describes the most extended techniques: *feature-based*, *direct* and *semi-direct* approaches. The chapter also analyzes the *remaining challenges* for visual odometry and SLAM while placing the work of this thesis in the context of visual SLAM.

**Chapter 3: Robustness to Low-textured Environments** describes the difficulties suffered by most traditional solutions in low-textured environments, in which their performance usually deteriorates. This chapter also presents four different thesis contributions, the first three are purely *odometry* approaches and the last one is *a complete SLAM system*, in the context of improving robustness of VO/SLAM in low-textured environments, all of them presented along with *publicly available C++ libraries*.

**Chapter 4: Dealing with Dynamic Illumination and HDR Environments** is one of the main *open challenges* in visual SLAM, where difficulties come from both the limitations of the sensors and the *bold assumptions* often introduced in the algorithms to alleviate computational requirements. In this chapter two different contributions to this topic are presented, one from a *deep learning* perspective, and another one from a *geometrical* point of view to allow feature tracking in such conditions.

**Chapter 5: Dataset for Low-textured, Dynamic Illumination and HDR Environments** proposes a visual-inertial dataset containing real-world visually challenging situations, focusing on environments with little texture or difficult and dynamic illuminations. The goal of this dataset is to provide a benchmark for the evaluation of visual-inertial odometry algorithms in these situations, allowing to evaluate the accumulated drift of the trajectory with the purpose of easily measuring and comparing results from different algorithms.

**Chapter 6: Conclusions** provides some final insights drawn from the work done in this thesis and briefly introduces the future lines still open to research in relation to the contributions of this work.



**Figure 1.2:** Scheme relating the main parts of this thesis.



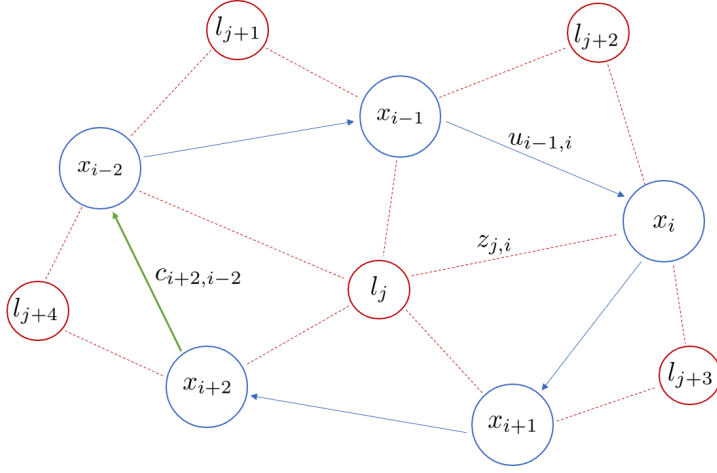
## Simultaneous Localization and Mapping

### 2.A Introduction

Typically, a modern SLAM system can be divided into two different parts, *i.e.* the *front-end* and *back-end*. The first one, the front-end, attempts to process the incoming data from on-board sensors and extracts relevant features, for instance in the case of images, distinguishable points from the environment. It also performs necessary *data association* between the features and their corresponding 3D landmarks, and loop closure detection or verification with the map data. In contrast, the back-end processes data from the front-end, hence inferring the state of the robot, while also estimating the optimal position of the 3D landmarks or sensor parameters, such as calibration, biases, and so on. Whereas it is fairly difficult to write analytically the front-end equations for any configuration, since it is highly dependent on the sensors mounted in the robot, the formulation for the SLAM back-end became standard in the early 2000s, especially after the surveys from Durrant-Whyte and Bailey in [16, 43].

Currently, the *de-facto* formulation of the SLAM back-end, was first introduced by Lu and Milios [99] followed by the work of Gutmann and Konolige [67]. Although a large number of approaches have been proposed to improve many aspects of the previous formulation, for instance robustness or accuracy, most of them coincide in posing the problem as *maximum a posteriori* (MAP) estimation. To explain this well-known formulation we have followed the notation in [24]. For that, let  $\mathcal{X}$  be the group of unknown variables defining the SLAM problem:

$$\mathcal{X} = \{x_i, l_j \mid (i \in 1, \dots, m), (j \in 1, \dots, n)\} \quad (2.1)$$



**Figure 2.1:** Typically, a SLAM problem is represented as a factor graph [39] where nodes represent robot states (red circles) and landmarks (blue circles), which are connected by sensor measurements (red arrows), odometry observations (blue arrows), or loop closures (green arrows).

with  $x_i$  and  $l_j$  typically representing the set of robot states and map landmarks, respectively. In order to estimate the solution to this problem, a number of measurements  $\mathcal{Z}$  is extracted from the specific sensor setting, also known as *observations*, i.e. :

$$\mathcal{Z} = \{z_k \mid (k \in 1, \dots, p)\} \quad (2.2)$$

where every measurement  $z_k$  is associated to a single state  $x_i$  and landmark  $l_j$  (for simplicity we omit these indices in the derivation) and can be expressed as a function, typically non-linear, of  $\mathcal{X}$ :

$$z_k = h_k(\mathcal{X}) + \epsilon_k \quad (2.3)$$

where  $h_k(\cdot)$  is known as the *observation model* (in practice, measurements only depend on a small subset of the variables), and  $\epsilon_k$  is a *random measurement noise*.

Then, the MAP estimation aims to obtain the value of  $\mathcal{X}$  by maximizing the *posterior*  $p(\mathcal{X}|\mathcal{Z})$ , i.e. , the *belief* over  $\mathcal{X}$  given the set of observations  $\mathcal{Z}$ , which can be expressed as follows after applying *Bayes theorem*:

$$\mathcal{X}^* = \underset{\mathcal{X}}{\operatorname{argmax}} p(\mathcal{X}|\mathcal{Z}) = \underset{\mathcal{X}}{\operatorname{argmax}} p(\mathcal{X}) p(\mathcal{Z}|\mathcal{X}) \quad (2.4)$$

where  $p(\mathcal{Z}|\mathcal{X})$  represents the *likelihood* of the observations given the set  $\mathcal{X}$ ,  $p(\mathcal{X})$  is the *prior* about  $\mathcal{X}$ , and  $\mathcal{X}^*$  is the optimal assignment that maximizes the posterior  $p(\mathcal{X}|\mathcal{Z})$ .

The MAP estimation in (2.4) equals to *maximum likelihood estimation* when there is no prior information about  $\mathcal{X}$ , since this term becomes constant and can be dropped from the optimization. In the case of assuming independent measurements  $\mathcal{Z}$ , it can be expressed as:

$$\mathcal{X}^* = \underset{\mathcal{X}}{\operatorname{argmax}} p(\mathcal{X}) \prod_{k=1}^p p(z_k|\mathcal{X}) \quad (2.5)$$

This problem is typically interpreted in terms of a *factor graph*, the variables  $\mathcal{X}$  corresponding to the nodes of the graph, where the *factors* of the graph represent probabilistic constraints to the graph over a subset of nodes. To illustrate this, the scheme depicted in Figure 2.1 shows the different nature of the factors typically considered in the SLAM problem.

The first type are the *odometry* factors, represented as  $u_{idx}$ , that typically constrain the motion from consecutive states, for instance incremental wheel odometry. Secondly, the sensor measurements can be represented by  $z_k$ , in this case it corresponds to the observation of the landmark  $l_j$  at the state  $x_i$ , and they encode useful information for the MAP estimation in (2.5), for instance, in the form of observations of a similar feature over several states. Finally, *loop closures* are represented by  $c_{idx}$ . Typically, they can be interpreted similarly to the above-mentioned factors, but with the particularity that they relate topologically distant nodes or states. In fact, this is one of the key aspects of a SLAM system. Without the inclusion of loop closure factors, the problem in (2.5) reduces to pure odometry since it interprets the world as an "infinite corridor" and in consequence, the system always considers the robot is exploring unseen areas. Alternatively, when including the loop closure factors the system recognizes that the new state,  $x_{i+2}$  in the case of Figure 2.1, belongs to a previously mapped area, which allows to reestablish the connections of the graph, hence reducing the accumulated drift over the robot trajectory.

If now it is assumed, as it is extended in literature, that the sensor measurements  $z_k$  follow Gaussian distributions with an expected value (mean) of  $h_k(\mathcal{X})$  and covariance matrix  $\Sigma_k = \Omega_k^{-1}$  its likelihood can be written as:

$$p(z_k|\mathcal{X}) \propto e^{-\frac{\|h_k(\mathcal{X}) - z_k\|_{\Omega_k^{-1}}^2}{2}} \quad (2.6)$$

where the exponent,  $\|h_k(\mathcal{X}) - z_k\|_{\Omega_k^{-1}}^2$ , is the Mahalanobis distance, typically:  $\|x\|_A^2 = x^T A x$ . Furthermore, under these circumstances, the MAP estimation

is equivalent to *minimizing the negative log-likelihood over the posterior* and hence, the problem in (2.5) can be expressed as:

$$\mathcal{X}^* = \underset{\mathcal{X}}{\operatorname{argmin}} -\log \left( p(\mathcal{X}) \prod_{k=1}^p p(z_k|\mathcal{X}) \right) = \underset{\mathcal{X}}{\operatorname{argmin}} \sum_{k=0}^p \|h_k(\mathcal{X}) - z_k\|_{\Omega_k}^2 \quad (2.7)$$

where it can be noticed that the prior has been written similarly to the sensor measurements for  $k=0$  (in practice, it is typically assumed to follow a uniform distribution thus neglected from the optimization). The previous formulation is for a versatile and well-known representation of the SLAM back-end, expression (2.7) is a *least squares* problem, hence allowing for many theoretical solutions, typically addressed for different *sparsity* levels of the graph.

In contrast, it is not straightforward to propose a unified formulation for the front-end part of a SLAM system since it strongly depends on the type of robot, the on-board sensors, or the application requirements. Initially, SLAM approaches were mainly proposed for mobile robots carrying 2D laser range-finders that commonly constrained the estimation to planar motions, and they also had an elevated price [40]. In that context, cameras started to be a highly suitable sensor choice for robotics applications due to their relatively cheap price in comparison with other sensors, and also since they proved to provide more useful information to other modules, such as obstacle avoidance, object recognition, task planning, semantic mapping, and a long etcetera. Although initially the proposed solutions were unfeasible for real applications, mainly due to the higher computational requirements, with the advances in computing the first *real-time* SLAM systems based on cameras started to appear [37] [78] and nowadays visual SLAM has reached an extraordinary maturity [24]. The rest of the chapter reviews the state of the art in visual SLAM regarding the most differentiable types of front-ends, and concludes with an overview of the open challenges for visual SLAM, putting in context the work of this thesis.

## 2.B Overview of Visual SLAM Techniques

Among others, visual SLAM (and odometry) has been addressed using a wide variety of sensors such as *monocular* [149], *stereo* [116], *RGB-D* [83], *omnidirectional* [28], and, more recently, *event* cameras [114] [56]. In regard to the type of errors utilized to tackle this problem with visual sensors, one can first differentiate between *indirect* and *direct* methods [49]. The former approach, better known as *feature-based*, typically pre-process the sensor measurements to produce an intermediate representation of the image information in form of features, commonly *keypoints*, then minimizing the *geometric errors* between the predictions and the observations. In contrast, the latter approach directly employs the actual sensor values, *i.e.* pixels intensity, over a time window, thus minimizing the *photometric errors*.

Recently, a hybrid approach, also known as *semi-direct*, has emerged with the aim to offer an efficient compromise between the two aforementioned approaches, for which they first use direct formulation for initial alignment and data association then refining the solution with the indirect formulation. In the following, we briefly describe these three different approaches in the current context.

### 2.B.1 Indirect Methods

In a nutshell, the front-end of a traditional feature-based approach has three main functionalities. The most important one provides accurate initial values to the problem in (2.7), typically by locally estimating the visual odometry through least-squares minimization of the geometric errors. The visual odometry problem is usually addressed in a similar way to the SLAM back-end, however, it only considers short-term data association and constraints, *i.e.*, it discards old parts from the map and loop closures. In fact, it can be seen as a simplified version of the problem in (2.7) where only local factors are considered, *i.e.* the map is never revisited. For instance, the simplest case would only estimate the current state of the robot simply considering observations from the previous frame hence dramatically reducing the computational requirements of the SLAM problem. In contrast, a more complex case would also consider several past states and the visible landmarks, in an intermediate approach between the full problem (2.7) and the simplest VO, hence requiring some mechanism to marginalize out redundant or past information to solve this problem in real-time, for instance considering only frames with common features with the current keyframe [115].

Accordingly, the front-end requires the implementation of a *data association* module which manages the tracking of the distinctive features, and also involves *outlier* detection and rejection, which require a pre-computation time. Moreover, the *depth* of the features needs to be estimated in order to initialize the 3D landmarks before recovering the different states. This step is addressed individually for every type of sensor since they require different levels of computation, for instance, while RGBD cameras allow to directly read the depth of a pixel, monocular techniques require several views and an explicit parametrization, *e.g.* *inverse depth* [31], and they also suffer from *scale ambiguity*.

Finally, the front-end also deals with long-term data association by detecting and validating previously visited places, *i.e.*, loop closures, and establishing correspondences with the detected part of the map [46]. The detection step is addressed by storing a database of encoded information for each keyframe, typically with a *bag-of-words* [57] approach employed to recognize previously visited places in the previous parts of the sequence. Then, the relative pose between the current and the previously visited keyframes can be estimated through the established correspondences, and then, the pose is further employed to correct the drift accumulated in the inner loop.

## 2.B.2 Direct Methods

One of the major drawbacks of feature-based techniques is the necessity of dealing with feature extraction and matching every incoming frame while also dealing with incorrect data association. This is a highly time consuming stage (it usually requires an elevated part of the per-frame available time) and consequently, in most algorithms this module is optimized for speed rather than precision. Moreover, since indirect methods rely on very distinctive geometric features, they only exploit a small part of the available information, which can bias the estimation if, for instance, the features are not well-distributed over the image.

In contrast, direct methods avoid to explicitly deal with this feature detection step by minimizing some *photometric error* based on the actual sensor readings from a certain location over a time period. As a result, such methods do not involve robust data association techniques since pixel are indirectly corresponded by the geometry of the problem, however, this usually requires a good initialization close to the convergence radius which is typically achieved with *coarse-to-fine* frameworks [50]. Although direct methods have proven to outperform feature-based methods in terms of robustness [49], they involve cumbersome operations to compute the photometric error, such as image warping over large regions of the image, often requiring GPU acceleration.

## 2.B.3 Semi-direct Approach

Alternatively, there are hybrid approaches, such as SVO [53], which benefit from the use of both the indirect and direct formulations. On one hand, semi-direct approaches usually involve feature detection and correspondence, however, the data association is an implicit consequence of direct image alignment. For that, semi-direct methods use small patches for every feature and obtains a rough estimation of the camera motion with this sparse model-based image alignment that, as a consequence, also provides feature correspondences. This is highly beneficial from a computational point of view, since feature detection is only performed whenever a new keyframe is inserted and there is no need of computing feature descriptors or matching them, and therefore, it highly reduces the computational time of the pipeline. On the other hand, as soon as the rough estimation of the camera pose and the feature correspondences are estimated, the algorithm then switches to the classic indirect formulation, which allows to efficiently solve the problem with the well-known indirect methods.

## 2.C Remaining Challenges for Visual Odometry and SLAM

In the last 20 years, visual SLAM techniques have reached maturity with impressive results achieved in controlled environments. In fact, the scientific community has considered SLAM as a theoretically solved problem since the past decade, of course in static environments, as Durrant-Whyte and Bailey stated in 2006 [43]: *At a theoretical and conceptual level, SLAM can now be considered a solved problem. However, substantial issues remain in practically realizing more general SLAM solutions and notably in building and using perceptually rich maps as part of a SLAM algorithm.*

Today, more than a decade later, its popularity has not stopped growing, it is still one of the most active research topics, and indeed, the question of *Is SLAM solved?* remains in the air for the scientific community [24]. The reason behind is that, even with the impressive results achieved by the state-of-art techniques, there are many open challenges to address before reaching a robust SLAM system, and also further research issues that deserve to be investigated.

One of the first remaining issues is the *robustness* of a SLAM system for *long-term* operations in uncontrolled environments, where classical assumptions do not stand anymore. For instance, it is typical to assume that the environment the robot is moving through remains *unchanged* and *static*, both in the short-term, *e.g.* , people surrounding the robot, and in the long-term, *e.g.* , the appearance of an office will inevitably change. Another phenomenon that particularly affects the accuracy of SLAM system is *perceptual aliasing*, which results in wrong *data association* due to both incorrect matches (*outliers*) or correct matches rejected by the front-end (*false negatives*). This thesis contributes to improve robustness in harsh or difficult environments, in particular:

**Low-textured Environments**, where it is usual that the performance of traditional approaches decreases due to difficulties in finding a sufficient number of reliable point features. The effect in such cases is an accuracy impoverishment and, occasionally, the complete failure of the system. In contrast, many of such low-textured environments contain planar elements that are rich in straight shapes, so an alternative feature choice, such as *line segments*, would exploit information from structured parts of the scene.

**Difficult Illumination** conditions or *high dynamic range (HDR)* environments. The main difficulties in these situations come from both the limitations of the sensors, *e.g.* , quick changes from dark to bright areas might over-expose the images, and the inability to perform a successful tracking of interest points because of bold assumptions in SLAM such as *brightness constancy*, which do not stand in these conditions.

Besides robustness, which has mainly focused the goals of this work, there are many other research topics that remain open from an academic and industrial point of view, such as *fail recovery*, robustness to *hardware failure*, *appearance changes* (such as day to night or seasonal ones), automatic *parameter tuning*, *scalability*, and a big etcetera. The reader can find a more complete review on the current state of SLAM and a more exhaustive analysis on some possible future research lines in [24, 36].

## Robustness to Low-textured Environments

### 3.A Introduction

The previous chapter briefly introduces the current techniques to address the Simultaneous Localization and Mapping (SLAM) problem, while at the same time it outlines the still open challenges to navigate through a robust system capable of working autonomously in uncontrolled situations. Typically, the performance of any of the traditional approaches already mentioned usually decreases in low-textured environments due to difficulties in finding a large or a fairly distributed set of keypoint features, which typically causes an accuracy impoverishment and even the complete failure of the system.

In contrast, most of such low-textured environments are rich in planar elements (they are typically human made indoor scenarios) which makes possible to extract informative line segment features from the linear shapes present in the scene. In this chapter, one of the main claims is that these two types of features, *i.e.* , keypoints and line segments, complement each other and its combination leads to a more versatile, robust and stable SLAM system capable of working in all types of scenarios. Additionally, the estimated maps comprising both types of 3D features provide a richer representation from the environment, thanks to the inclusion of structural information from the scenario. As a consequence, a number of applications performing higher level tasks (such as place recognition, semantic mapping, task planning, etc.) can significantly benefit from the useful information that can be inferred from the combined maps.

### 3.B Contributions

In this context, the work of this thesis first contribute in [62] with a complete probabilistic stereo visual odometry system that, thanks to the combination of both points and line segments, is capable of robustly working in such difficult environments. Unfortunately, the difficulties in dealing with line segment features involves, among others, a higher complexity in both the representation and the computational requirements for its detection and tracking, hence increasing the complexity of this problem. Moreover, the contribution presented in [123] employed this stereo visual odometry system to develop and test a robust probabilistic model for the projection errors of point features based on real data by modeling them with Gamma distributions which improved both precision and accuracy of the system.

To alleviate this additional requirements, in the second contribution of this chapter [61] a popular semi-direct approach to monocular visual odometry, known as SVO [53], to simultaneously exploit information from line segments. In consequence, the proposed system allowed for a faster feature tracking, since the semi-direct framework eliminated the necessity of continuously extracting and matching features between subsequent frames, and for a more robust system capable of dealing with both textured and structured environments.

At last, the contributions of this chapter conclude with PL-SLAM [65], a real-time stereo visual SLAM system that combined both points and line segments to work robustly in a wider variety of scenarios. In this work, the impact of both type of features is leveraged at all instances of the SLAM pipeline: visual odometry, keyframe selection, bundle adjustment, loop closing, etc. Additionally, the resulting map is richer and more diverse in 3D elements, which can be exploited to infer valuable, high-level scene structures like planes, empty spaces, ground plane, etc.

The developed algorithms have been compared with other state-of-art solutions in well-known and publicly available datasets and benchmarks. Additionally, open-source C++ implementations of the proposed algorithms were released along with the published articles and some extra multimedia material for the benefit of the community.

---

## 3.C Robust Stereo Visual Odometry through a Probabilistic Combination of Points and Line Segments

---

Ruben Gomez-Ojeda and Javier Gonzalez-Jimenez

*Published in Proc. International Conference on Robotics and Automation  
(ICRA), 2016.*

©IEEE (Revised layout)

# Robust Stereo Visual Odometry through a Probabilistic Combination of Points and Line Segments

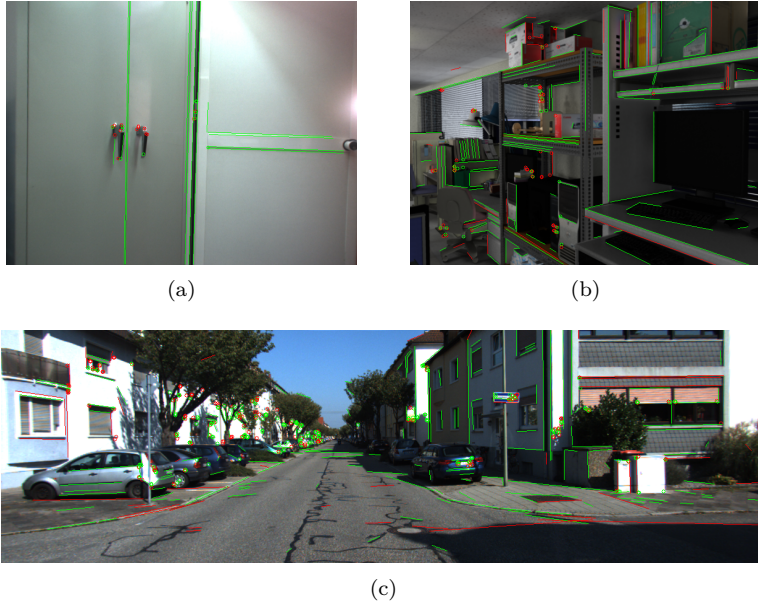
*Ruben Gomez-Ojeda and Javier Gonzalez-Jimenez*

## Abstract

Most approaches to stereo visual odometry reconstruct the motion based on the tracking of point features along a sequence of images. However, in low-textured scenes it is often difficult to encounter a large set of point features, or it may happen that they are not well distributed over the image, so that the behavior of these algorithms deteriorates. This paper proposes a probabilistic approach to stereo visual odometry based on the combination of both point and line segment that works robustly in a wide variety of scenarios. The camera motion is recovered through non-linear minimization of the projection errors of both point and line segment features. In order to effectively combine both types of features, their associated errors are weighted according to their covariance matrices, computed from the propagation of Gaussian distribution errors in the sensor measurements. The method, of course, is computationally more expensive than using only one type of feature, but still can run in real-time on a standard computer and provides interesting advantages, including a straightforward integration into any probabilistic framework commonly employed in mobile robotics.

### 3.C.1 Introduction

In recent years, visual odometry (VO) has gained importance in robotics applications such as ground vehicles moving on uneven terrains, or unmanned aerial vehicles (UAVs). An alternative to VO in these cases is the use of inertial measurement units (IMUs), but they are not able to cancel the gravity effects precisely, accumulating large errors over time. Traditional solutions also include wheel odometry, which cannot replace VO since it only works with smooth and planar movements, and GPS-based navigation systems, which are limited to open outdoor environments and are unable to estimate the orientation of the device they are attached to. An additional advantage of VO is that the information required (provided by cameras) can be exploited for other navigation-related tasks such SLAM [113] and scene recognition. Visual odometry can be addressed with a single camera [45] [87] [51], stereo cameras [86], or RGB-D sensors [84] [76]. Moreover, two methodologies have been considered



**Figure 3.1:** (a) Texture-less scenes are challenging for traditional point-based SVO approaches. (b) Synthetic frame extracted from the Tsukuba dataset. (c) Frame extracted from the KITTI dataset, where both point and line features are abundant.

in the literature: appearance-based and feature-based. The first group, known as dense approach, works on the whole image assuming some kind of photo-consistency between the successive frames [53] [50]. An alternative strategy consists of matching some relevant features (either points or lines) in the images, and then estimates the pose increments by establishing some rigid-body constraints between those features. Most visual odometry systems are based on feature points, since they are easily detectable and matchable. Some remarkable works following this approach are [119] and [59]. In the former, the authors report a stereo visual odometry (SVO) system based on an iterative estimation of the 6DoF camera motion. The point features are detected with a variant of the Harris corner detector and matched according to their normalized correlation. In the latter, the authors propose an algorithm which employs point features in combination with a sparse feature matcher to reconstruct the 3D pose of a stereo camera given a sequence of images. Those methods have proven to work fast and robustly in many environments, but their behavior in low textured scenes, such as the one in Figure 3.1(a), deteriorates since it is difficult to find a large set of reliable points. In contrast, line segments are usually abundant in any human-made environment, even in low textured scenes, but these methods are not so common in literature since the detec-

tion of lines involves a high computational cost. In that context, Witt and Weltin [150] proposed the Iterative Closest Multiple Lines (ICML) algorithm, where the Iterative Closest Point (ICP) algorithm is adapted to the case of line segments. This approach estimates simultaneously the correct matches and the pose increment by considering one-to-many line matches inside a non-linear optimization process, which works well under the assumption of small rotations. While this proposal yields a good performance in fast video sequences, it has certain tendency to fall into local minima, thus the authors also propose a robust hypothesize-and-test algorithm as a failure detection step. However, in highly textured environments (e.g. outdoor scenes) the number and quality of the detected lines decreases, and, in consequence, the performance of the algorithm. This problem is addressed by Koletschka et al. [90] with a strategy that efficiently combines point and line features, and hence it can work in different environments. They also propose an algorithm for the stereo matching of the line segments which computes the sub-pixel disparity of the endpoints of the line and deals with partial occlusions. None of the above-mentioned proposals takes into account the probabilistic entity of the features employed since they face the SVO problem in a deterministic way. While this alternative has the advantage of being more efficient computationally, the probabilistic treatment of the variables reduces the undesirable effect of noisy measurements in the optimization, and allows the estimated variables (poses and landmarks) to be easily integrated in probabilistic frameworks which are commonly used in mobile robotics. In this paper we propose a complete probabilistic SVO system that works robustly in different environments thanks to the combination of both points and line segments, which usually provide complementary information. The incremental pose of the stereo camera is recovered iteratively through probabilistic on-manifold optimization of the projection errors, which are computed between the projected features from the first frame and those detected in the second frame. We estimate the uncertainty of all the variables involved in the stereo process, which are assumed to follow Gaussian distributions, and then introduce them as weights in the cost function minimization, increasing the robustness to noise and yielding more accurate results. The source code of the developed C++ Stereo Visual Odometry library is available online, and will be updated as research progresses. An illustrative video of our SVO system and the source code can be found here: <http://mapir.isa.uma.es>

### 3.C.2 System Overview

In a nutshell, we track the features (points and segments) in a sequence of stereo frames and compute their 3D position and their associated uncertainty. The 3D landmarks are then projected to the new camera pose, where an error function is minimized in order to end up with both the pose increment of the camera and the uncertainty of this estimation. In the following we introduce

each step of the SVO system and describe the most important details of its implementation.

### Point Features

For dealing with feature points, we employ the ORB [126] detector and descriptor due to its efficiency and good performance. In order to reduce the number of outliers, we only consider the measurements that are mutual best matches, i.e. the best match in the left image corresponds to the best match in the right one. To ensure that the correspondences are meaningful enough, we also check that the distance in the description space between the two closest matches is above certain threshold, which is set to the double of the distance of the best match. We also ensure a fair distributions of points over the input images with a bucketing approach that divides the image in 16 buckets, and tries to add at least 20 features in each.

### Line Segment Features

The line segments are detected with the Line Segment Detector (LSD) [147], which has a high precision and repeatability. However, it is time consuming, which is its major weakness for real-time applications. To mitigate this, we detect the line segments in a parallelized framework in both stereo images. For the stereo matching and frame-to-frame tracking we first compute the LBD descriptors [154] for each line, and match them based on their local appearance features. Similarly to the case of points, we check that both features are mutually best matches, and also that the best two matches are sufficiently separated in the description space. We have not applied a bucketing strategy in the line segments detection, since it provides less reliable features and hence yields poorer results.

### Motion Estimation

Once the features have been tracked from a stereo frame to the next one, the line segment endpoints and the feature points are back-projected. Then, the motion is estimated iteratively through a probabilistic Gauss-Newton minimization of the line and point projection errors. The negative effect of incorrect correspondences is reduced by employing a Pseudo-Huber loss function to detect and remove the outliers, as proposed in [112]. The complete process will be detailed in Section 3.C.3.

### Uncertainty Propagation

In order to improve the precision of the incremental pose estimation, we weight the errors with the inverse of the error covariance matrix. This covariance matrix is obtained by propagating the feature errors which are assumed to be

zero-mean Gaussian distributed (a common hypothesis in computer vision). Ultimately, this propagation process ends up with the uncertainty of the estimated pose, which makes our system suitable to be easily integrated in any probabilistic robotic algorithm. The error distributions will be described and validated in Section 3.C.4.

### 3.C.3 Combined Stereo Visual Odometry

The straightforward approach to compute the camera motion in a SVO system minimizes the error of the 3D features reconstructed from two consecutive stereo frames (i.e. 3D error minimization as in [111]). This procedure has the advantage of a closed form solution but, in practice, it is not the best option since it is strongly affected by the euclidean errors induced by the noisy measurements of the features, which may lead to large motion error in the estimated odometry. Instead, a more precise approach is that of projecting the 3D points (the endpoints in the case of line features) from the first frame to the second one, thus the motion is obtained by 2D error minimization of the features in the image.

#### 3.C.3.1 Problem Statement

Let  $C$  and  $C'$  be the stereo coordinate systems (typically placed at the left camera) at two consecutive poses, related by the relative transformation  $\mathbf{T}(\boldsymbol{\xi}) \in SE(3)$ , where  $\boldsymbol{\xi} \in \mathfrak{se}(3)$  is the 6-vector of coordinates in the Lie algebra  $\mathfrak{se}(3)$ . The problem we face is that of estimating the optimal  $\mathbf{T}(\boldsymbol{\xi}^*)$  that minimizes the projection error for points and line segments (expressions (3.1) and (3.2) below, respectively) under the hypothesis that the measurements are affected by unbiased Gaussian noise (as modeled in Section 3.C.4). The stereo camera is assumed to be in an ideal configuration with a baseline  $b$ , and the calibration parameters  $\mathbf{K}$  are either provided by the manufacturer or known from previous calibration. The point projection error  $\Delta \mathbf{p}_i(\boldsymbol{\xi})$  is given by:

$$\Delta \mathbf{p}_i(\boldsymbol{\xi}) = \hat{\mathbf{p}}_i(\boldsymbol{\xi}) - \mathbf{p}'_i \quad (3.1)$$

with  $\mathbf{p}'_i$  being the  $i$ -th detected point in the second frame, and  $\hat{\mathbf{p}}_i(\boldsymbol{\xi})$  the projected point from the first frame to the second one, both in homogeneous coordinates. With that notation, we define the line equation in the second image  $\mathbf{l}'_j$  as the cross product between the endpoints of the line in homogeneous coordinates, denoted as  $\mathbf{p}'_j$  and  $\mathbf{q}'_j$  respectively. The line projection error is defined as a vector formed by the euclidean distances from the projected endpoints of the line segments in the first frame and the line detected in the second frame, i.e.:

$$\Delta \mathbf{l}_j(\boldsymbol{\xi}) = [\mathbf{l}'_j]^\top \cdot [\hat{\mathbf{p}}_j(\boldsymbol{\xi}) \ \hat{\mathbf{q}}_j(\boldsymbol{\xi})]^\top \quad (3.2)$$

where  $\hat{\mathbf{p}}_j(\boldsymbol{\xi})$  and  $\hat{\mathbf{q}}_j(\boldsymbol{\xi})$  refer to the projected endpoints, and  $\mathbf{l}'_j$  is the  $j$ -th infinite line detected in the second frame.

### 3.C.3.2 On-Manifold Optimization

The optimal pose increment  $\mathbf{T}(\boldsymbol{\xi}^*)$  is computed through an iterative minimization of the Maximum Likelihood Estimator (MLE) which selects the model  $\boldsymbol{\xi}^*$  for which the probability of the observed data becomes maximum. Under the assumption that the data is corrupted by unbiased Gaussian noise, the MLE coincides with the following non-linear least-squares estimator:

$$\boldsymbol{\xi}^* = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \left\{ \sum_i^{N_p} \Delta \mathbf{p}_i(\boldsymbol{\xi})^\top \boldsymbol{\Sigma}_{\Delta \mathbf{p}_i}^{-1} \Delta \mathbf{p}_i(\boldsymbol{\xi}) + \sum_j^{N_l} \Delta \mathbf{l}_j(\boldsymbol{\xi})^\top \boldsymbol{\Sigma}_{\Delta \mathbf{l}_j}^{-1} \Delta \mathbf{l}_j(\boldsymbol{\xi})^\top \right\} \quad (3.3)$$

where  $N_p$  and  $N_l$  corresponds to the number of point and line correspondences respectively, and the matrices  $\boldsymbol{\Sigma}_{\Delta \mathbf{p}_i}^{-1}$  and  $\boldsymbol{\Sigma}_{\Delta \mathbf{l}_i}^{-1}$  are the  $2 \times 2$  inverse of the covariance matrices for each type of feature. We calculate the optimal solution through iterative Gauss-Newton optimization on the manifold tangent space  $\mathfrak{se}(3)$ . In this case, the Jacobian matrix is expressed as follows:

$$\mathbf{J}(\boldsymbol{\xi}) = \left. \frac{\partial \mathbf{E}(\boldsymbol{\xi} \oplus \boldsymbol{\varepsilon})}{\partial \boldsymbol{\varepsilon}} \right|_{\boldsymbol{\varepsilon}=\mathbf{0}} \quad (3.4)$$

where the vector  $\mathbf{E}$  contains both line and point projection errors, and the operator  $\oplus : \mathfrak{se}(3) \times \mathfrak{se}(3) \mapsto \mathfrak{se}(3)$  is a generalization of the normal addition operator for Euclidean spaces. For further details on the mathematics, please refer to [19].

### 3.C.3.3 Fast Outlier Rejection

Due to inaccuracies in the feature detection and tracking process the presence of outliers in the observed data is unavoidable, which leads the optimization process to unreliable results. Besides, the assumption of Gaussian distribution errors renders the system to be highly vulnerable to outliers. In order to deal with this phenomena we have implemented a variant of the ERODE outlier detector [112], which performs a fast and efficient outlier removal based on radial distributions, due to its computational performance. Concretely, we have employed the Cauchy loss function to robustify the MLE:

$$\rho(s) = \log(1 + s) \quad (3.5)$$

where the input  $s$  corresponds to each component of the error vector in (3.3). With this function, the minimization process converges to the true solution, and after a few iterations, the outliers can be easily detected and removed as they present large residuals. Finally, the minimization process is relaunched with the inliers to obtain the optimal solution.

### 3.C.4 Uncertainty of the Error Functions

The advantage of combining different types of features of the scene, namely points and line segments, relies on their proper weighting in the cost function, which in turn comes from their observation errors. Specifically, this is implemented in the optimization process by weighting the measurements with the inverse of the uncertainty of the projection error from each feature, as expressed in (3.3) with the matrices  $\Sigma_{\Delta \mathbf{p}_i}^{-1}$  and  $\Sigma_{\Delta \mathbf{l}_j}^{-1}$ . These matrices, which are intended to account for errors in image quantization and in the detection process, are obtained by estimating the Jacobians of the error functions (equations (3.1) and (3.2)) with respect to the observations  $\mathbf{x}$ , which includes both point and line segment observations  $\mathbf{p}_i$  and  $\mathbf{l}_j$  respectively, i.e.:

$$\Sigma_{\Delta k_i} \approx \frac{\partial \Delta k_i}{\partial \mathbf{x}} \Sigma_{\mathbf{x}} \frac{\partial \Delta k_i}{\partial \mathbf{x}}^\top \quad (3.6)$$

where the subindex  $k \in \{\mathbf{p}, \mathbf{l}\}$  refers to the type of the error function (points or lines). The observation uncertainties  $\Sigma_{\mathbf{x}}$  are modeled as bi-dimensional Gaussians with standard deviations  $\sigma_x = \sigma_y = 1$  pixel in the image plane, for both points and endpoints of the line segments. As stated in [70], the uncertainty of the optimal pose is approximated by the inverse of the Hessian of the cost function in (3.3), expressed as

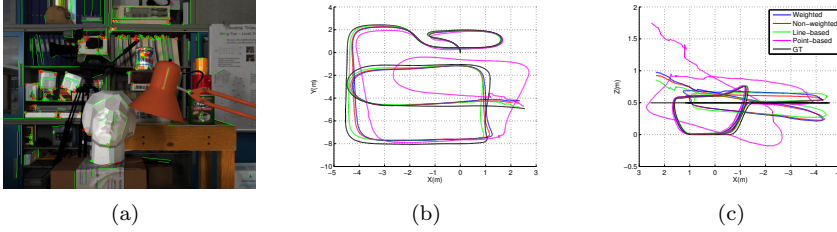
$$\Sigma_{\xi^*} \approx (\mathbf{J}(\xi^*)^\top \mathbf{W}(\xi^*) \mathbf{J}(\xi^*))^{-1} \quad (3.7)$$

where  $\mathbf{J}(\xi^*)$  is the full Jacobian in (3.4) that contains both point and line error functions, and  $\mathbf{W}(\xi^*)$  is a block-diagonal matrix containing the uncertainty of each projection error for each type of feature. Then, the camera pose increment follows a 6D normal distribution with mean the optimal pose  $\xi^*$ , and covariance matrix  $\Sigma_{\xi^*}$

$$\xi \sim \mathcal{N}(\xi^*, \Sigma_{\xi^*}) \quad (3.8)$$

#### 3.C.4.1 Detecting ill-Pose Configurations

For some spatial distributions the problem may be ill-posed. Such situations can not be detected before the optimization process, since it also depends on the relative motion of the camera. However, this information can be derived from the covariance matrix  $\Sigma_{\xi^*}$ . If we express this matrix in diagonal form, their elements give us the variance of the estimated motion parameters  $\xi^*$  in the space of the eigenvectors. This information can be employed to neglect those motion terms whose uncertainty is too high. This strategy is very useful when data from other sources, such as IMUs, GPS, wheel odometry, etc., are available, and that information can be fused with our SVO estimation leading to more robust solutions.



**Figure 3.2:** New University of Tsukuba Stereo Dataset. (a) Line and point matches after visual odometry estimation (the outliers and inliers are plotted in red and green respectively). (b) Top view of the trajectory. (c) Side view of the trajectory.

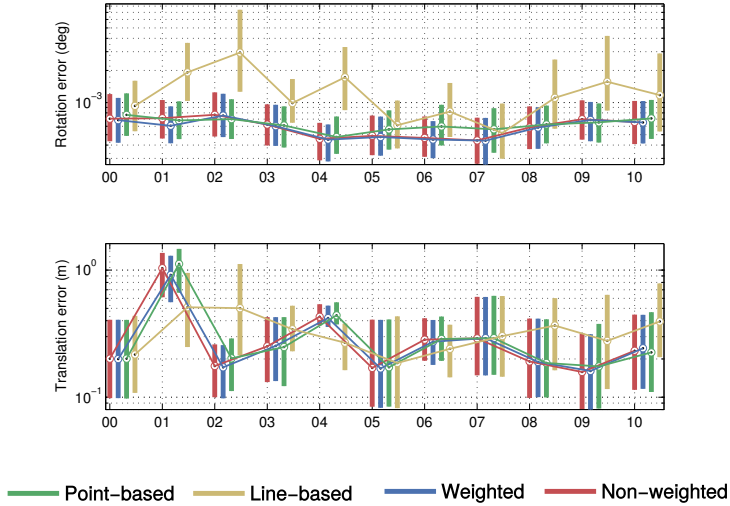
### 3.C.5 Experimental Validation

In this section we illustrate the benefits of the weighted combination of points and segments. For that, we estimate the trajectory of a stereo camera in several video sequences acquired in different environments.

#### 3.C.5.1 Video Sequences

##### Tsukuba dataset

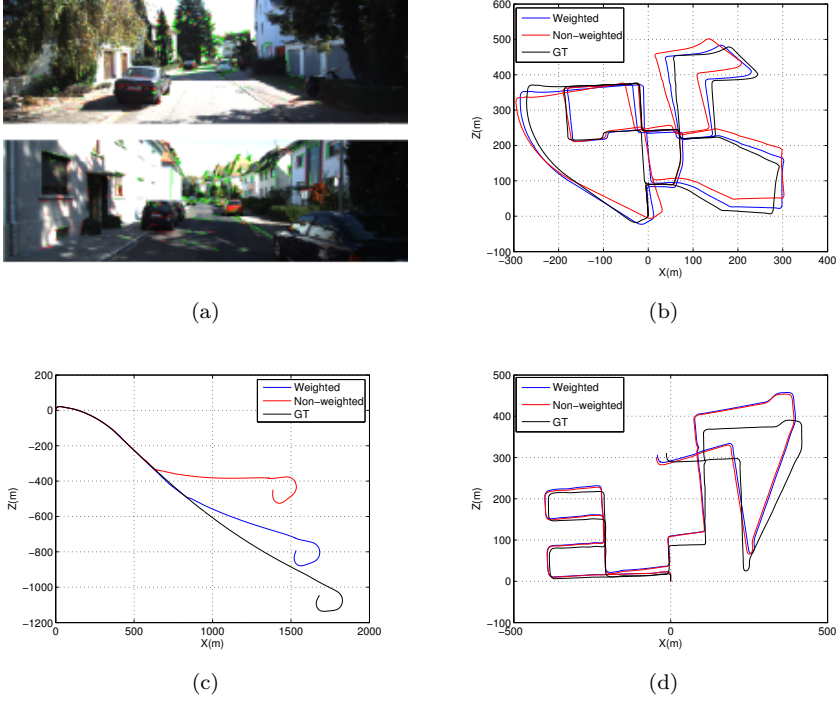
In this experiment we employ the New University of Tsukuba Stereo Dataset [120] (Figure 3.2(a)), which contains 1800 stereo pairs from a synthetic laboratory scenario for different illumination conditions. The stereo camera performs a 3D trajectory (about 50 meters length) over the laboratory scene, with several changes in orientation. We compare the accuracy of our SVO system, with the one that does not weight the measurements, and also check the advantages of combining point and line features by representing the trajectories obtained with one type of feature. Figures 3.2(b) and 3.2(c) depict both the top and side view of the estimated trajectories with the ground truth. During most of the sequence both estimations (the weighted and non-weighted) show a high accuracy, however, the non-weighted method presents a small superiority since this is a noise-free synthetic dataset, and therefore the uncertainty of the measurements is almost negligible. In the final part of the scene there is a door which induces a lateral drift into the non-weighted method, while the weighted trajectory keeps smooth. The reason for that is an increase in the number of bad measurements, whose negative effect in the quality of the estimation is avoided thanks to the employment of the uncertainty and also the Pseudo-Huber loss function in the process of detecting and removing the outliers. We also observe a superior behavior of the line-based algorithm with respect to the point-based, but obviously the solution which employs both features presents a better performance since that combination provides more information to the system which increases its accuracy.



**Figure 3.3:** RPE distributions of the SVO system in the KITTI dataset sequences, comparing the performance of the combined weighted (in blue) and non-weighted (in red) methods, and also the point (in green) and line-based (in yellow) methods.

### KITTI dataset

We also have used the KITTI benchmark [58], which provides accurate ground truth based on a Velodyne laser scanner and a GPS localization system. The stereo camera rig is formed by two gray-scale Point Grey Flea<sup>®</sup>2 video cameras separated with a baseline of 54 cm attached to the top of a car. As already mentioned, the introduction of proper weights for the different features in the optimization improves the accuracy of the estimated trajectory, since it limits the influence of those landmarks with high uncertainty. For checking that, we compare the results obtained with our strategy, that weights the features with their uncertainty, with one that does not employ this information (non-weighted approach). We also compare it with implementations that consider only one type of feature: either points or line segments. Figure 3.3 plots the distributions of the relative pose errors (RPE) [140] of the rotation and translation components between all camera pose increments for the test sequences of the KITTI dataset. This chart confirms the superior performance of both combined methods, which works well in most scenes while the behavior of both point and line-based systems is irregular since they are more influenced by the structure of each environment. In general, the point-based approach is superior to the line-based in the KITTI dataset, since it is a highly textured dataset where most lines found do not provide enough information to recover the 6D pose. It also can be noticed a slight out-performance of the weighted



**Figure 3.4:** Sequences extracted from the KITTI benchmark. (a) Line and point correspondences from two different frames after filtering geometrically inconsistent matches. (b) Top view of the KITTI-00 trajectory. The ground truth is represented with black lines, the estimation weighted with the uncertainty with blue lines, and the non-weighted estimation is plotted with red lines. (c) Top view of the KITTI-01 trajectory. (d) Top view of the KITTI-08 trajectory.

method. However, the major benefits of including the uncertainty in the optimization process can be observed visually in Figure 3.4. The top views of the 3D estimated trajectories and the ground truth are represented in Figures 3.4(b) and 3.4(c)Figure 3.4(d), while a frame from the scene is plotted in Figure 3.4(a). We observe a good performance of the weighted method during the three sequences, while the non-weighted algorithm suffers from a big error in the rotation estimation of the sequence KITTI-01 that deviates the trajectory from the ground truth. This is caused by a series of noisy measurements during the medium part of the sequence, that induces the non-weighted method to a poor estimation of the camera motion while the weighted methods, even those which employ only one type of feature, are capable of inhibit the influence of these bad landmarks thanks to the uncertainty weighting.

### 3.C.5.2 Comparison in the KITTI Vision Benchmark

In this section we compare the performance of our SVO method with several state-of-art algorithms in the evaluation sequences from the KITTI dataset. A deeper comparison would test the performance of our method against those in [150] and [90] in various environments, since both approaches employ line segment features in the pose estimation, however, we have not found any public implementation of them. Table 3.1 shows the results of several feature-based VO algorithms, as reported in the KITTI benchmark website, which unlike previous experiments it measures the accumulated trajectory error. Although the performance of our method is slightly inferior in terms of relative translation errors, with an error of 3.26% against the 2.44% of VISO2S [59], and relative rotation errors, with errors of 0.0095 deg/m in comparison with the 0.0077 deg/m of TGVO [86], its main advantage is the robust performance in noisy and low-textured scenarios, when most point-based methods usually fails [150].

**Table 3.1:** Comparison of several VO systems in the KITTI Vision Benchmark.

Method	Tran.(%)	Rot.(deg/m)	Time(s)
Ours	3.26	0.0095	0.20
VISO2S [59]	2.44	0.0114	0.05
TGVO [86]	2.94	0.0077	0.06
VO3ptLBA [12]	3.13	0.0104	0.57
VISO2M+GP [59] [137]	7.46	0.0245	0.15
VISO2M [59]	11.94	0.0234	0.10

### 3.C.5.3 Processing Time

In this section we first compare the average execution times of both weighted and non-weighted approaches. Table 3.2 shows the average computation times and number of correspondences per frame, for the stereo sequences with different resolutions. It can be noticed a slight increment in the execution time of the weighted approach due to the computation of the weights, which can be perfectly assumed by most applications in mobile robotics. We also analyze the influence of the image resolution in the processing time. First, we observe that our stereo visual odometry system runs in average with frequencies of 12 Hz for  $640 \times 480$  images, with a high number of correspondences processed (an average number of 96 lines and 78 points). The proposed SVO system can work with frequencies superior to 30 Hz when the resolution is set to lower values ( $320 \times 240$ ). In that case, since the average number of detected line correspondences decreases, the accuracy of the camera pose estimation may drop.

**Table 3.2:** Average number of correspondences and processing times per frame.

Dataset	Resolution	Lines	Points	Weighted		Non-weighted	
				Frequency	Runtime	Frequency	Runtime
Tsukuba	$320 \times 240$	45	40	31.40 Hz	31.85 ms	32.01 Hz	31.24 ms
Tsukuba	$640 \times 480$	96	78	12.04 Hz	83.05 ms	12.38 Hz	80.76 ms
KITTI	$613 \times 185$	54	60	21.47 Hz	46.57 ms	22.00 Hz	45.46 ms
KITTI	$1226 \times 370$	75	188	4.54 Hz	220.03 ms	4.57 Hz	218.99 ms

### 3.C.6 Conclusions

In this paper we have introduced a novel stereo visual odometry system based on points and line features, thus capable of working in different environments. For effectively combining them we take into account the uncertainty of the measurements, which improves the accuracy of the estimation. Besides, the probabilistic distribution of the poses provided by our algorithm can be implemented in any probabilistic framework commonly adopted in robotic applications. In addition, we have confirmed the theoretical results through a series of real experiments in both synthetic and real environments, by estimating the trajectory of different stereo cameras. Future work will focus on improving the performance of our SVO system by introducing a different weighting that will reduce the impact that bad measurements or mobile objects has in the algorithm.

---

## 3.D Accurate Stereo Visual Odometry with Gamma Distributions

---

Ruben Gomez-Ojeda, Francisco-Angel Moreno, and Javier Gonzalez-Jimenez

*Published in Proc. International Conference on Robotics and Automation  
(ICRA), 2017.*

©IEEE (Revised layout)

# Accurate Stereo Visual Odometry with Gamma Distributions

Ruben Gomez-Ojeda, Francisco-Angel Moreno, and Javier Gonzalez-Jimenez

## Abstract

Point-based stereo visual odometry systems typically estimate the camera motion by minimizing a cost function of the projection residuals between consecutive frames. Under some mild assumptions, such minimization is equivalent to maximizing the probability of the measured residuals given a certain pose change, for which a suitable model of the error distribution (sensor model) becomes of capital importance in order to obtain accurate results. This paper proposes a robust probabilistic model for projection errors, based on real world data. For that, we argue that projection distances follow Gamma distributions, and hence, the introduction of these models in a probabilistic formulation of the motion estimation process increases both precision and accuracy. Our approach has been validated through a series of experiments with both synthetic and real data, revealing an improvement in accuracy while not increasing the computational burden.

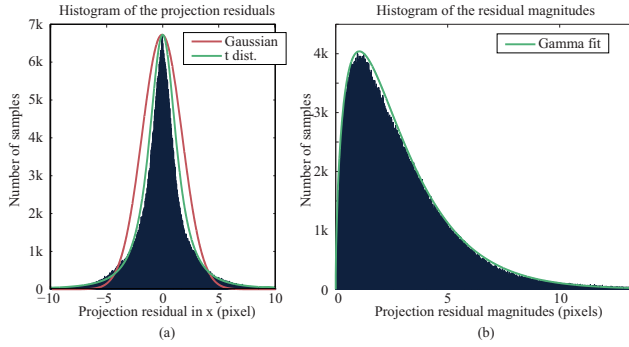
### 3.D.1 Introduction

Most stereo visual odometry systems estimate camera motion through the least-squares minimization [44, 55] of a certain cost function  $C(\xi)$  of the residuals  $\Delta \mathbf{p}_i(\xi)$ , defined as difference between the *observations*  $\mathbf{p}'_i$  of a set of keypoints and their *predictions*  $\hat{\mathbf{p}}_i(\xi)$  [60]:

$$\Delta \mathbf{p}_i(\xi) = \hat{\mathbf{p}}_i(\xi) - \mathbf{p}'_i, \quad (3.9)$$

where  $\hat{\mathbf{p}}_i(\xi)$  is computed by back-projecting to 3D the observed  $i$ -th image keypoint detected in the previous frame, and then re-projecting it to the current one, according to an estimation of the pose change  $\xi \in \mathfrak{se}(3)$  between them. Typically, the cost function is derived from the maximization of the probability of the pose change given the residuals, so that minimizing  $C(\xi)$  (i.e. estimating the optimal pose change  $\xi^*$ ) is equivalent to maximizing  $p(\xi | \Delta \mathbf{p})$ . This is also equivalent to maximizing their likelihood given a certain pose change, under the assumptions of independent and equally distributed noise, and a uniform *prior* distribution over the poses:

$$\xi^* = \underset{\xi}{\operatorname{argmax}} p(\xi | \Delta \mathbf{p}) = \underset{\xi}{\operatorname{argmax}} p(\Delta \mathbf{p} | \xi). \quad (3.10)$$



**Figure 3.5:** Histogram of (left) the projection residuals in the  $x$ -coordinate for the keypoints extracted from the sequence KITTI00, and the corresponding fitted Student’s  $t$ -distribution (in green) and Gaussian distribution (in red), and (right) the residual magnitudes of the keypoints extracted from all the training sequences in the KITTI dataset. The real distribution can be described accurately with a Gamma distribution.

In this context, finding a proper model of the residual distribution becomes of capital importance as the results will be directly affected by its goodness of fit. Furthermore, such model should consider the presence of not only noise but also outliers.

Commonly, keypoint predictions (and consequently the projection residuals) are considered to be Gaussian-distributed, since the observed keypoints are assumed to be corrupted by Gaussian noise that is propagated through linear approximations of the above-mentioned back-projection and re-projection functions [33]. In practice, though, such approximations still present inaccuracies, as can be seen in Figure 3.5(a), which depicts the distribution of the residuals in the image  $x$ -coordinate computed from real data (which is similar to that of the  $y$ -coordinate). As a consequence, assuming a Gaussian distribution for  $p(\Delta \mathbf{p} | \boldsymbol{\xi})$  leads to an unsuitable cost function whose minimization will yield inaccurate results. In fact, according to the real distribution, a better approach would be to model the residual in both the  $x$  and  $y$  image coordinates by a Student’s  $t$ -distribution, whose shape is similar to the Gaussian one but with heavier tails (refer again to Figure 3.5(a)). This approach has been explored in [84] and applied to RGB-D cameras.

Nevertheless, we propose to employ the *magnitude* of the residual between observations and predictions  $\mathbf{r} = \{r_i(\boldsymbol{\xi}) = \|\Delta \mathbf{p}_i(\boldsymbol{\xi})\|\}$ , instead of the projection residual  $\Delta \mathbf{p}$ . Thus, we claim that modeling  $\mathbf{r}$  as a Gamma distribution (i.e.  $\mathbf{r} \sim \Gamma(\theta, \alpha)$ ) is a better option than modeling residuals as either a Gaussian or a  $t$ -distribution, since the fitted model deviate less from the actual distribution

it approximates (see Figure 3.5(b)). Following this, and the above-mentioned assumptions, the optimization problem in (3.10) becomes:

$$\xi^* = \underset{\xi}{\operatorname{argmax}} p(\xi | \mathbf{r}) = \underset{\xi}{\operatorname{argmax}} p(\mathbf{r} | \xi). \quad (3.11)$$

The introduction of the Gamma distribution in the optimization process allows us to derive a more suitable cost function that leads to better results than assuming that residuals follow either a Gaussian or a t-distribution, as will be proved with a series of experiments.

The two parameters of such Gamma distribution (namely shape and scale) are estimated at each time-step from the actual histogram of all the involved residual magnitudes, being necessary a minimum number of samples for the fit to be representative. The on-line fitting procedure introduces little additional cost to the optimization process while the benefits are two-fold: a more precise camera pose estimation and more robustness against outliers and noisy measurements than the standard Gaussian-based approach. Even so, a very large ratio of outliers may eventually degrade its performance, so the usage of *robustification* methods is still advisable.

Our claim is supported by an extensive experimental validation with both synthetic and real data, revealing its suitability for performing visual odometry, specially for stereo vision systems where observations in both images can be employed. For that, our proposal has been integrated into our previous stereo visual odometry (SVO) system presented in [62]. The results show significant improvements in accuracy whilst incurring in a reduced computational footprint. An illustrative video of our system and the SVO library source code can be accessed in <http://mapir.uma.es/rgomez>.

### 3.D.2 Related Work

Visual-based motion estimation algorithms are strongly affected by the presence of noisy data and, specially, outliers, which do not follow the commonly assumed Gaussian distribution for the residuals, hence eventually leading the system to erroneous results. Traditional approaches to this problem, as the one in [86], often rely on variants of RANSAC to deal with wrong measurements by generating a solution which is in consensus with the majority of the dataset. However, this technique has high computational requirements. In [121], Person et al. presented a stereo visual odometry system which takes advantage of monocular techniques, as they argue that those techniques are more refined and robust than those of stereo systems. For that, they implement a delayed outlier identification procedure based on an essential matrix RANSAC approach and robust iterative triangulation. Other approaches, as the one in [53], integrate robust probabilistic filters to explicitly deal with outliers by estimating, for instance, the depth at feature locations over multiple frames. Then, these depth filters are updated at each frame labeling as inliers

those points with low uncertainty in depth, hence being introduced into the map and subsequently employed to estimate the camera motion.

Finally, other approaches introduce some robust cost functions in the camera motion estimation, hence obtaining appropriate weights that reduces the impact of wrong measurements. A first group proposes several modifications of the well-known extended Kalman filter (EKF) in order to increase the robustness of their systems against outliers and noisy measurements. In [143] authors propose a robust EKF filter to deal with outliers in real-time, by down-weighting the samples with more probability of being outliers, for which they learn the system dynamics thus avoiding manual parameter tuning. In [10] the previous approach was generalized and extended by introducing efficient smoothing and filtering modifications for dealing with data corrupted with non-Gaussian and heavy-tailed noise. The previous work was also extended in [11], where authors proposed to introduce a structured variational approximation with a more robust and flexible behavior, and yet introducing only a little increment in the computational complexity. Another group of techniques model directly the error distribution, and then perform a robust non-linear least-squares minimization of these errors. In [84], Kerl et al. perform robust odometry estimation for RGB-D cameras by minimizing the photometric error between two consecutive frames. They argue that their dense RGB-D residuals can be better explained with Student's  $t$ -distributions, for which they derive a probabilistic formulation including a robust sensor model based on real world data. Recently, the work in [9] proposes a generic self-tuning M-estimator which iteratively estimates the parameters of the residual distribution, thus removing the necessity of manually set such parameters. However, this method needs to compute the importance weights for each iteration of the least-squares problem, hence being computationally expensive for small problems as the one we address here.

### 3.D.3 Distribution of the Projection Errors and Residuals

In this section, we empirically analyze the actual distributions of both the projection residual  $p(\Delta \mathbf{p} | \xi)$  and the residual magnitude  $p(r | \xi)$  for the case of image keypoints. For this purpose, we detect and match ORB keypoints along a sequence of stereo images provided by the KITTI collections of public datasets [58]. Then, the observed keypoints are projected to the next frame by applying the ground truth pose increment (also included in the dataset), and both the residuals and their magnitude are computed. Finally, we adjust different distributions to the data and evaluate their goodness of fit.

Regarding the projection residuals, we refer again to Figure 3.5(a), which has been built from the sequence "00" of the KITTI dataset. As stated before, it can be seen that the residual in the  $x$  image coordinate (and similarly for the  $y$  coordinate) does not follow a Gaussian distribution. In fact, these data can be more properly fitted by a  $t$ -distribution, as pointed out in [84]. So, we

**Table 3.3:** Average goodness of fit with the K-S test for each distribution, with a critical value of 0.0608 for  $\alpha = 0.05$ 

Seq.	Frames	Feats	Proj. Residual		Magnitude Gamma
			Gaussian	t-dist.	
00	4540	777k	0.1455	0.0827	<b>0.0474</b>
01	1100	100k	0.2327	0.1447	<b>0.0591</b>
02	4660	1059k	0.1035	0.0638	<b>0.0474</b>
03	800	200k	0.1222	0.1518	<b>0.0475</b>
04	270	55k	<b>0.0533</b>	0.1199	<b>0.0532</b>
05	2760	478k	0.1023	0.0725	<b>0.0492</b>
06	1100	133k	0.1673	0.1256	<b>0.0499</b>
07	1100	199k	0.1790	0.1313	<b>0.0464</b>
08	4070	731k	0.1412	0.0916	<b>0.0456</b>
09	1590	273k	0.1429	0.0745	<b>0.0481</b>
10	1200	191k	0.1298	<b>0.0534</b>	<b>0.0478</b>

may consider to use this distribution to derive a suitable cost function that takes into account a better approximation of the residual true distribution. However, we claim that modeling the residual magnitude as a Gamma distribution instead of the residual as either a Gaussian or a t-student represents a more accurate fit of the modeled variable.

To prove this, we also analyze the distribution of the residual magnitude, shown in Figure 3.5(b), where all the training sequences in the KITTI dataset have been employed to build the histogram. It can be observed that a Gamma distribution accurately describes the behavior of the magnitude, as it presents a certain bias and also a heavy tail. The goodness of the three fits (i.e. Gaussian and t-distribution for the projection residual and Gamma for the residual magnitude) are evaluated through the Kolmogorov-Smirnov (K-S) test [138], which measures the maximal difference between an empirical and a real distribution function. Thus, for each sequence, a subset of  $10^3$  keypoints has been randomly selected from all the found features so that half of them are employed to derive the distribution model, while the rest is used to perform the test. Note that using separate datasets is mandatory in order to obtain valid and distribution-free K-S test results [15], hence allowing the comparison of different distributions. This experiment has been repeated  $10^3$  times for each sequence, obtaining the average values shown in Table 3.3. In all sequences, the values below the test's critical value have been highlighted (which is 0.0608 for a significance value of  $\alpha = 0.05$ ). As expected, the t-distribution approach approximates better the real distribution of the residual than the Gaussian model. Nonetheless, the results also reveal that the Gamma distribution represents a more accurate model for the residual magnitude than the

t-distribution for the projection residual in most datasets. Then, modeling their magnitude as a Gamma distribution (and consequently deriving a cost function of the residual magnitude according to that) will lead to more accurate results than employing a cost function of the projection residuals based on the t-distribution.

Finally, it is important to remark that the number of samples (i.e. observed keypoints) employed to fit the distributions influences the quality of the approximation, as will be further discussed in Section 3.D.5.

### 3.D.4 Motion Estimation with the Gamma Distribution

In this section, we derive the equations to robustly recover the 6D pose change  $\xi$  of a stereo camera using the Gamma-based approach to model the behavior of the residual magnitude. For that, let us formally define the vector of residual magnitudes  $\mathbf{r}(\xi) = \{r_i(\xi)\}$  that contains the projection distances of all the individual observations, as defined in Section 3.D.1. Then, we aim to find the camera motion  $\xi^* \in \mathfrak{se}(3)$  that maximizes the posterior probability  $p(\xi | \mathbf{r})$  as stated in equation (3.11), which we reproduce here for clarity:

$$\xi^* = \underset{\xi}{\operatorname{argmax}} p(\xi | \mathbf{r}) = \underset{\xi}{\operatorname{argmax}} p(\mathbf{r} | \xi). \quad (3.12)$$

Under the mild assumptions of  $r_i(\xi)$  being independent, estimating (3.12) is equivalent to minimizing the negative log-likelihood of the residual magnitude (refer to [82, 141] for further details):

$$\xi^* = \underset{\xi}{\operatorname{argmax}} p(\mathbf{r} | \xi) = \underset{\xi}{\operatorname{argmin}} \left\{ - \sum_i \log p(r_i | \xi) \right\} \quad (3.13)$$

Now, we model the magnitude  $\mathbf{r}(\xi)$  with a Gamma distribution, i.e.  $\mathbf{r} \sim \Gamma(\alpha, \theta)$ , whose probability density function (pdf) is given by:

$$f(x; \alpha, \theta) = \frac{1}{\Gamma(\alpha)\theta^\alpha} x^{\alpha-1} e^{-x/\theta} \text{ for } x > 0 \text{ and } \alpha, \theta > 0 \quad (3.14)$$

where  $\alpha$  and  $\theta$  are the so-called shape and scale parameters, respectively. Then, the individual likelihood of the residual magnitude is proportional to:

$$p(r_i | \xi) \propto r_i^{\alpha-1} e^{-r_i/\theta} \quad (3.15)$$

where we have dropped the constant terms that do not depend on  $\xi$ . Finally, by introducing this model into (3.13), the estimator becomes:

$$\xi^* = \underset{\xi}{\operatorname{argmin}} \sum_i \left\{ r_i/\theta - (\alpha - 1) \log r_i \right\} \quad (3.16)$$

which is equivalent to minimizing this cost function (following an Iteratively Re-Weighted Least Squares (IRLS) approach):

$$\xi^* = \underset{\xi}{\operatorname{argmin}} \sum_i w(r_i(\xi)) r_i^2(\xi) \quad (3.17)$$

with  $w(r_i(\xi))$  being a weighting function defined by:

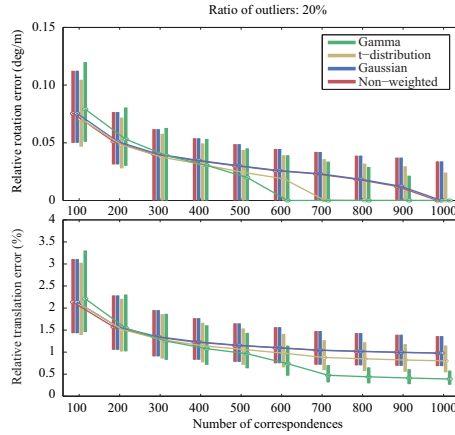
$$w(r_i(\xi)) = \frac{r_i/\theta - (\alpha - 1)\log r_i}{r_i^2}. \quad (3.18)$$

For the sake of computational complexity, we fit the Gamma distribution at each time-step with the Method of Momentums, which employs the closed form solutions for both the mean  $\mu = \alpha\theta$ , and the variance of the distribution  $\sigma^2 = \alpha\theta^2$  (a complete comparative of several methods for fitting Gamma distributions can be found in [32]). However, using these parameters entails that outliers also have influence in the estimation of both the mean and variance. Therefore, it is desirable to employ robust methods to estimate the distribution parameters [73]. Thus, we employ the Median Absolute Deviation for the standard deviation  $\hat{\sigma}$  and, subsequently, we estimate a robust mean  $\hat{\mu}$  by only considering the samples lying less than three times  $\hat{\sigma}$ .

Once the Gamma distribution has been fitted, we derive the weighted cost function in equation (3.17), which is optimized on the  $\mathfrak{se}(3)$  manifold through the well-known Gauss-Newton equations (refer to [19] for a thorough analysis on on-manifold optimizations). Finally, although our proposed cost function presents some robustness against outliers due to the sublinear nature in the residual magnitude of the weighting equation (3.18), it is important to remark that a big amount of them still may degrade the resulting ego-motion estimation. Therefore, we have implemented a variant of the ERODE outlier detector [112] with the main difference of using a Cauchy distribution instead of a Huber loss function and employed as an outlier-removal strategy for all the tested approaches in the experiments.

### 3.D.5 Experimental Evaluation

This section presents two sets of experiments that analyze the effect in the localization accuracy of introducing the proposed Gamma-based model in our robust SVO system [62] in comparison to other approaches, namely: i) non-weighted, ii) Gaussian-weighted, and iii) Student's t-distributed weighted. For the first approach, we perform a standard least-squares minimization of the residuals without defining any weight for them. For the last two, we fit a Gaussian or a t-distribution to the computed projection residuals, respectively, and derive a cost function from equation (3.13), which will define the weights for the individual residuals.



**Figure 3.6:** Rotation (top) and translation (bottom) errors over a variable number of observations, employing different cost functions: non-weighted (in red), Gaussian weighted (in blue), Student’s t-distribution (in yellow), and Gamma distribution (in green).

### 3.D.5.1 Experiments with synthetic data

In this first set of experiments, we have generated random stereo observations (keypoints) in two consecutive frames, related by a random camera motion. Thus, image keypoints are randomly spread all over the first stereo pair, by simulating the point locations in the left image as well as their corresponding disparities. Then, we project them to the current stereo frame according to a random camera motion and, subsequently, Gaussian distributed noise is added to each keypoint in both stereo frames. Finally, we compute the motion estimation error in different scenarios.

In these experiments, we have simulated camera motions that follow a uniform distribution between  $\pm 1$  m and  $\pm 3$  deg, which emulates a camera moving at similar speeds to those presented in [58]. The disparity of stereo points has been set to follow a uniform distribution between [10,30] pixels, while the camera intrinsic parameters are those specified for the KITTI dataset.

### Impact of the Number of Observations

As discussed in Section 3.D.1, the number of keypoint correspondences has a strong influence in the quality of the fitted Gamma distribution. To assess this, we have evaluated our SVO approach for a variable number of observations through the following Monte-Carlo simulation: for each weighting method and number of observations, we estimate the camera pose change for 1000 different configurations of both observations and camera motions, resulting in 1

million simulations. The outliers ratio has been set to 20 % in this series of experiments.

Figure 3.6 plots the results for the evaluated methods, where we have measured both rotation and translation average errors (along with 95% confidence intervals, plotted as solid bars), specified in deg/m and % of the total length, respectively, with respect to the true camera motion. As expected, both rotation and translation errors show a slightly superior performance of the other three methods in comparison to our approach for the lowest number of observations, since there is not enough information to fit a Gamma distribution properly. In contrast, this tendency is inverted as the number of observations increases, revealing our method to clearly outperform the other three approaches in both precision and accuracy, specially over 600 landmarks.

### Impact of the Ratio of Outliers

Now, we study the impact of the number of outliers in the accuracy of the camera motion estimation, keeping a fixed number of 200 keypoints. Again, we have performed 1 million simulations for 1000 different configurations of observations and camera motions, respectively.

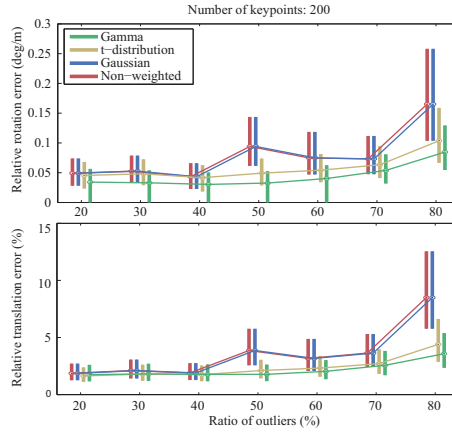
The results are depicted in Figure 3.7, where it is shown a clearly better performance of both Gamma and t-distribution against the non-weighted and Gaussian-weighted approaches, since the first ones present a robust behavior (as discussed in previous section). Gamma and t-distribution approaches performs similarly in both translation and rotation errors for lower ratios of outliers, although the Gamma-based sensor model provide more accurate results when increasing the number of outliers, since they can be easily detected and removed from the residual magnitude distribution.

#### 3.D.5.2 SVO Evaluation in the KITTI Benchmark

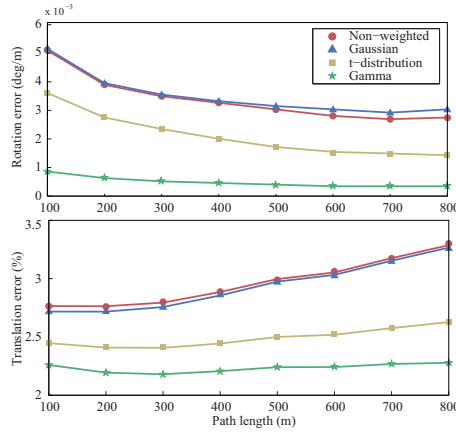
In this section, we assess the impact of the different approaches when performing robust camera ego-motion estimation for the training sequences ("00" to "10") from the KITTI dataset [58]. For that purpose, we have evaluated the results by using again the same metrics employed in the KITTI Benchmark, which computes errors in both rotation and translation for different subsequences lengths and speeds.

#### Performance at Different Sequence Lengths

First, we compute both rotation and translation errors relative to the distance traversed, for all the different subsequence lengths considered in the dataset



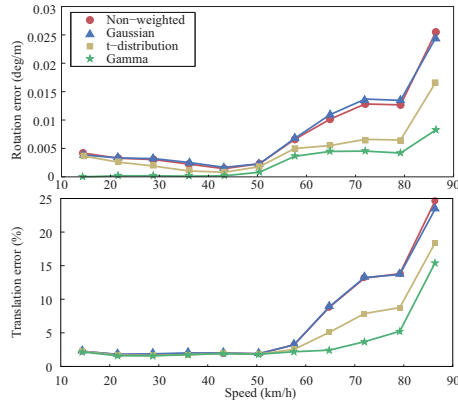
**Figure 3.7:** Rotation (top) and translation (bottom) errors over all sub-sequences of a given length in the KITTI dataset, employing different cost functions: non-weighted (in red), Gaussian weighted (in blue), Student’s t-distribution (in yellow), and Gamma distribution (in green).



**Figure 3.8:** Average rotation (top) and translation (bottom) errors over all sub-sequences of a given length in the KITTI dataset, employing different cost functions: non-weighted (in red circles), Gaussian weighted (in blue triangles), Student’s t-distribution (in yellow squares), and Gamma distribution (in green stars).

(100m, 200m,..., 800m). The results show a significant improvement in both errors for all the different subsequences with our method, which performs clearly better than the rest of the approaches (refer to Figure 3.8).

Although the t-distributed weighting scheme also improves the ego-motion estimation in comparison to the Gaussian-weighted approach, our proposal



**Figure 3.9:** Average rotation (top) and translation (bottom) errors over all subsequences of a given speed in the KITTI dataset, for the different cost functions considered.

yields better results since it describes more accurately the actual nature of the residual magnitude distribution. Moreover, it can be seen that the relative improvement of our approach, in comparison with the rest, grows as the path length increases. This is caused, in part, by the good performance obtained regarding to rotations, since high errors in rotation deviate the absolute trajectory from the ground truth, hence increasing absolute translational errors.

### Performance at Different Speeds

In this experiment, we analyze the impact of the different weighting functions when performing visual odometry for all the speeds considered in the KITTI Benchmark (4m/s, 6m/s, ..., 24m/s), by computing again the average rotation and translation errors (refer to Figure 3.9). It can be seen that our proposal clearly presents a superior performance for all the considered speeds, specially over 60km/h. This is caused by an increasing number of wrong measurements and outliers introduced to the system when the camera is traveling at high speeds, due to difficulties in feature tracking (those sequences usually correspond to low-textured highway scenes). In these situations, our proposed Gamma-based model performs better than the rest of the methods since it describes the actual nature of the residual magnitude distribution, so that outliers and wrong measurements are down-weighted properly, as claimed in this paper.

**Table 3.4:** Average optimization time per frame for a given number of observations.

#Observations	nWeight.	Gauss.	t-dist.	Gamma
$N \leq 200$	0.911 ms	1.034 ms	1.134 ms	1.113 ms
$200 < N \leq 300$	1.399 ms	1.625 ms	1.783 ms	1.748 ms
$300 < N \leq 400$	1.872 ms	2.212 ms	2.420 ms	2.378 ms
$400 < N \leq 500$	2.329 ms	2.787 ms	3.038 ms	2.991 ms
$N > 500$	2.962 ms	3.590 ms	3.899 ms	3.854 ms

### Computational Time

Finally, we analyze the computation time employed by each algorithm in the optimization process, for all the frames in the training set of the KITTI dataset. Experiments have been conducted on a single core of an Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz processor with 4GB RAM. Table 3.4 contains the average time per frame employed for each algorithm, for a given number of observations.

As expected, the non-weighted approach has the lowest computational footprint, as it does not involve any weight estimation. On the other hand, although the Gaussian weighted approach requires less computational cost than the Gamma-weighted and t-distributed weighted approaches, it performs similar to the non-weighted approach (as demonstrated in the previous experiments), thus not justifying its application. Finally, our proposal slightly outperforms the t-distributed weighting scheme, with a smaller computational burden, while increasing the accuracy in visual odometry estimation. Hence, it is interesting to consider the inclusion of the proposed Gamma-based model in robust systems with higher requirements in accuracy than in computational time.

### 3.D.6 Conclusions

In this paper we have proposed a Gamma-based model for the distribution of the projection residual magnitude in keypoint-based stereo-visual odometry. This approach is employed to derive a proper cost function of the residual magnitude which accurately weights each individual observation according to their true distribution. Its minimization leads to a robust ego-motion estimation that outperforms other weighting approaches that model projection errors as Gaussian or Student's t-distributions. Moreover, our proposal also presents robustness against outliers, since the model reproduces the tail behavior of the residual magnitude real distribution so that outliers are properly down-weighted in the optimization process. The claimed features have been proved with extensive visual odometry experiments with both synthetic and real data,

where we compare our approach with the non-weighted, Gaussian-distributed, and t-distributed approaches.

---

## 3.E PL-SVO: Semi-Direct Monocular Visual Odometry by Combining Points and Line Segments

---

Ruben Gomez-Ojeda, Jesus Briales, and Javier Gonzalez-Jimenez

*Published in Proc. International Conference on Intelligent Robots and Systems (IROS), 2016.*

©IEEE/RSJ (Revised layout)

# PL-SVO: Semi-Direct Monocular Visual Odometry by Combining Points and Line Segments

*Ruben Gomez-Ojeda, Jesus Briales, and Javier Gonzalez-Jimenez*

## Abstract

Most approaches to visual odometry estimates the camera motion based on point features, consequently, their performance deteriorates in low-textured scenes where it is difficult to find a reliable set of them. This paper extends a popular semi-direct approach to monocular visual odometry known as SVO [53] to work with line segments, hence obtaining a more robust system capable of dealing with both textured and structured environments. The proposed odometry system allows for the fast tracking of line segments since it eliminates the necessity of continuously extracting and matching features between subsequent frames. The method, of course, has a higher computational burden than the original SVO, but it still runs with frequencies of 60Hz on a personal computer while performing robustly in a wider variety of scenarios.

### 3.E.1 Introduction

Visual odometry (VO) is gaining importance in robotic applications, such as unmanned aerial vehicles (UAVs) or autonomous cars, as an essential part of the navigation systems. Solutions for the VO problem has been addressed employing different sensors, such as monocular or stereo cameras [87] [47] [121], RGB-D cameras [84] [76], or a combination of any of them with an Inertial Measurement Unit (IMU) [52]. The traditional approach consist of the detection and matching of point features between frames, and then, the estimation of the camera motion through least-squares minimization of the reprojection errors between the observed and projected points [128]. In this context, the performance of such approaches deteriorates in low textured scenarios as depicted in Figure 3.10, where it is difficult to find a large or well-distributed set of image features. In contrast, line segments are usually abundant in human-made scenarios, which are characterized by regular structures rich in edges and linear shapes. Dealing with line segments in images it is not as straightforward as points, since they are difficult to represent [18] and also require high computational burden for the detection and matching tasks thus only a few solutions have been proposed [29] [90], barely reaching real-time specifications. Moreover, edge-based algorithms have been also used for both solving the problem



**Figure 3.10:** In low-textured environments, point-based algorithms usually fail due to difficulties in founding a large number of features, in contrast, line segments are usually abundant.

of tracking [69] [144] [125], and estimating the camera motion [92]. However, these methods require a rather costly direct alignment which makes them less suitable to real time, and also limits their application to narrow baseline estimations. To the best of our knowledge, this paper proposes the first real-time approach to Monocular Visual Odometry (MVO) that integrates both point and line segment features, and hence it is capable of working robustly in both structured and textured scenarios. The source code of the developed C++ PL-SVO library and illustrative videos of this proposal can be found here: <http://mapir.isa.uma.es>

### 3.E.1.1 Related Work

Visual odometry algorithms can be divided into two main groups. The first one, known as *feature-based*, extract a set of image features (traditionally points) and track them along the successive frames. Then, they estimate the pose by minimizing the projection errors between the correspondent observed features and those projected from different frames. Literature offers us several point-based approaches to the odometry problem, such as PTAM [88], where authors report a fast SLAM system capable of performing real-time parallel tracking and mapping over thousands of landmarks. In contrast, the problem of motion estimation with line features has been less explored due to their inherent difficulties, specially to monocular odometry. In [150] authors extend the Iterative Closest Point (ICP) approach [98] to the case of stereo odometry with line segments, where they substitute the computation of costly descriptors in a one-to-multiple line matching approach. In our previous work [62], we present a stereo visual odometry system based on both point and line segment features. The influence of each feature is weighted with the inverse of their covariance matrix, which is obtained by uncertainty propagation techniques

over the reprojection functions. However, this work still relies on traditional feature detection and matching, and thus it has a high computational cost.

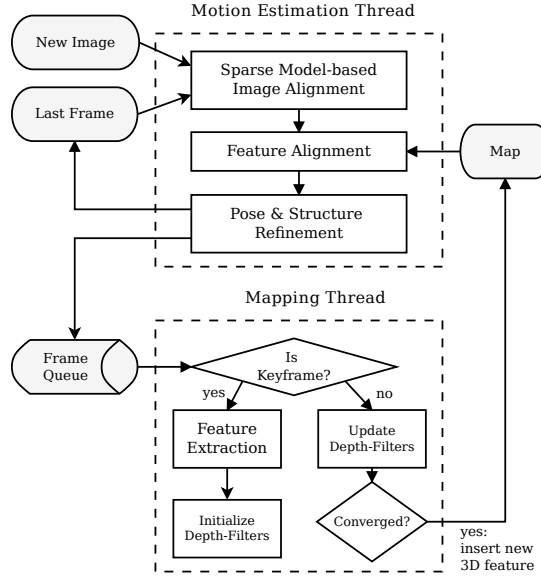
The other group, known as *direct* approaches, estimates the camera motion by minimizing the photometric errors between successive frames at several image locations. In [118] authors propose a direct approach, known as DTAM, where they estimate the camera pose by direct alignment of the complete intensity image between each keyframe, employing a dense depth-map estimation. However, this method requires GPU parallelization since it process the whole image. For dealing with the high computational requirements of direct methods, a novel monocular technique is proposed in [51], where authors estimate the camera motion in a semi-dense approach, thus reaching real-time performance on a CPU. They continuously estimate and track a semi-dense inverse depth-map for regions with a sufficient image gradient, thus only exploiting those areas which introduce valid information to the system. Then, they estimate the camera motion by minimizing the photometric error over the regions of interest, hence combining the good properties of direct algorithms with a sparse approach that allows for fast processing.

### 3.E.1.2 Contributions

Point features are less abundant in low-textured and structured scenarios, and hence, the robustness and accuracy of point-based visual odometry algorithm dramatically decreases. On the other hand, detecting and matching line segments demands high computational resources, which is the main reason of the lack of real-time approaches to visual odometry using these features. In this work, we extend the semi-direct approach in [53] to the case of line segments, which performs fast feature tracking as an implicit result of a sparse-direct motion estimation. Therefore, we take advantage of this sparse structure to eliminate costly segment detection (we only detect them when a new keyframe is introduced to the framework), and descriptor computation, while maintaining the good properties of line segments. As a result, we contribute with a fast monocular odometry system capable of working robustly in low textured scenarios thanks to the combination of the information of both points and segment features. In the following we describe the proposed system, and validate the claimed features with experiments in different environments.

### 3.E.2 System Overview

The proposed system can be understood as an extension of the semi-direct framework in [53], that not only consider points but also segment features in the scene and introduce both in the pipeline. This is a non-trivial extension, since line segments present more complexity than point features from a geometrical point of view. In practice, this makes that certain image operations which are almost trivial for points become more computationally cumbersome



**Figure 3.11:** SVO framework, extracted from [53]. Our work extends the concept of feature so that both points and segments in the scene are considered for every step in the pipeline.

for the case of segments. Hence, we need to perform several approximations and take some well-founded heuristic in order to save computational resources. These will be seen in higher detail in the Sections 3.E.3 and 3.E.4.

For the sake of completeness, we briefly review every stage of the semi-direct framework [53], depicted in Figure 3.11 while showing how the partnering of this semi-direct approach and the use segment features becomes mutually beneficial. The semi-direct approach is divided into two parallel threads, one for estimating the camera motion and another one for mapping the environment.

### 3.E.2.1 Motion Estimation

In the motion thread, an initial motion estimate is performed between consecutive frames by using a sparse direct alignment approach (see Figure 3.12), which minimizes the photometric error between patches using the 3D warping provided by the known 3D features. This allows for the fast tracking of features between frames as a result of the semi-direct motion estimation, which eliminates the need of performing frame-to-frame detection and matching. This, which is fairly advantageous for point features, becomes extremely beneficial for segments since they are considerably more computationally expensive. Instead, features are only detected when a new keyframe is inserted, so that the overall cost of the LSD segments detection [147] becomes affordable. Further-

more, by reducing the dimension of the optimization problem to the estimation of the pose only epipolar geometry is automatically fulfilled and we do not have to take care of outlier matches.

Then, the second step (see Figure 3.13) of the motion thread is to refine the feature projections given by the transformation estimate from direct alignment, thus violating the epipolar constraints to reduce the drift of the camera. The feature refinement is performed by taking as reference patch the one with the closest viewpoint. This approach, again, is very beneficial for segments since it limits large observations baselines during the tracking of the segments. In consequence, it alleviates well-known issues of line segments such as endpoints repeatability, occlusions or deformation of the segments due to change of view [90]. Finally, both the camera motion and the map structure are refined by minimizing the reprojection errors.

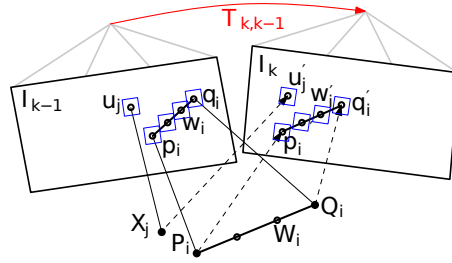
At this point, the matching with far features is fully solved thanks to the intermediate continuous tracking and we can apply specific feature-based refinement approaches that behave quite well for segment features, as depicted in Figure 3.14.

To sum up, we see that the introduction of segments in a semi-direct framework [53] can be done much more seamlessly than for other more traditional approaches, since the preliminary direct steps alleviate most of the downsides that have historically prevented the use of segments in Visual Odometry. Otherwise stated, the motion as well as the mapping are enriched by the use of segments without incurring in a significant overhead of the overall system.

Concurrently, the map thread estimates the depth of 2D features with a probabilistic Bayesian filter, which is initialized when a keyframe is inserted to the pipeline. The depth filters are initialized with a high uncertainty, but they converge to the actual values in a few iterations and are then inserted to the map, becoming useful for motion estimation. In the following, we describe in detail each stage of the algorithm, and then validate it with experiments in real environments.

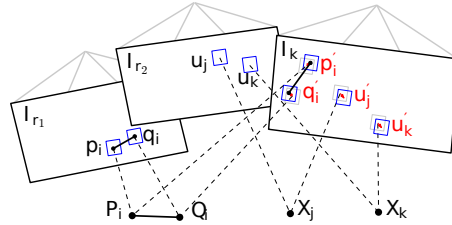
### 3.E.3 Semi-Direct Monocular Visual Odometry

Let  $C_{k-1}$  and  $C_k$  be the coordinate systems of a calibrated camera at two consecutive poses, which are related by the relative pose transformation  $\mathbf{T}_{k-1,k}(\boldsymbol{\xi}) \in SE(3)$ , where  $\boldsymbol{\xi} \in \mathfrak{se}(3)$  is the 6-vector of coordinates in the Lie algebra  $\mathfrak{se}(3)$ . The problem we face is that of estimating the camera pose along a sequence of frames, for which we denote  $\mathbf{T}_{k,w}$  as the camera pose with respect to the world's reference system in the  $k$ -th timestep. For that, let us denote as  $I_k$  the intensity image in the  $k$ -th frame, and  $\Omega$  as the image domain. We will denote the point features as  $\mathbf{x}$ , and its correspondent depth as  $d_{\mathbf{x}}$ . In the case of line segments, we will employ both the endpoints, denoted by  $\mathbf{p}$  and  $\mathbf{q}$  respectively, and the line equation as  $\mathbf{l}$ . The 3D point back-projected from the image at timestep  $k$  is denoted as  $\mathbf{X}_k$ , and can be obtained through the inverse



**Figure 3.12:** The relative pose between the current and the previous frame parameterizes the position of the reprojected points in the new image. We perform (sparse) image alignment to find the pose that minimizes the photometric difference between image patches corresponding to the same 3D point (blue squares). For the segments points are homogeneously sampled between the 3D endpoints. Note, in all figures, the parameters to optimize are drawn in red and the optimization cost is highlighted in blue. This figure has been adapted from [53].

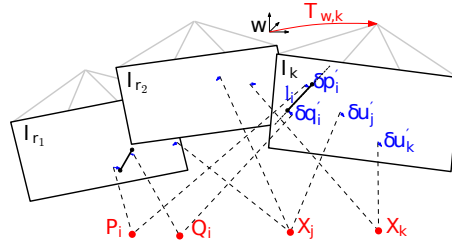
projection function  $\pi^{-1}$ , i.e.  $\mathbf{X}_k = \pi^{-1}(\mathbf{x}, d_{\mathbf{x}})$ . The projection of a 3D point in the image domain is obtained through the camera projection model  $\pi$ , so that  $\mathbf{x} = \pi(\mathbf{X}_k)$ . In the following, we will extend the steps of SVO algorithm to the case of line segments.



**Figure 3.13:** The 2D position of each point is optimized individually to minimize the photometric error in its patch. For the segments the end points are similarly optimized. This alleviates errors propagated from map and camera pose estimation. This figure was also adapted from [53].

### 3.E.3.1 Sparse Model-based Image Alignment

The camera motion between two consecutive frames,  $\mathbf{T}_{k-1,k}(\xi)$ , is first estimated through direct image alignment of the sparse features tracked along the frames. Unlike the point-based approach, we cannot directly align the whole region occupied by line segment between two frames, since it would be computationally expensive. For that, we only minimize the image residuals between some patches equally distributed all along the line segment, as depicted in



**Figure 3.14:** In the last motion estimation step, the camera pose and the structure (3D points and segments) are optimized to minimize the reprojection error that has been established during the previous feature-alignment step. Similarly to the previous ones, this figure has been adapted from [53].

Figure 3.12. Let us define  $\mathcal{L}$  as the image region for which the depth of the endpoints is known at previous time step  $k-1$ , and for which the endpoints  $\mathbf{p}$  and  $\mathbf{q}$  are visible in the image domain at the current timestep  $\Omega_k$ :

$$\begin{aligned} \mathcal{L} := & \{ \mathbf{p}, \mathbf{q}, \mathbf{w}_n \mid \mathbf{p}, \mathbf{q} \in \mathcal{L}_{k-1} \\ & \wedge \pi(\mathbf{T}(\boldsymbol{\xi}) \cdot \boldsymbol{\pi}^{-1}(\mathbf{p}, d_{\mathbf{p}})) \in \Omega_k \\ & \wedge \pi(\mathbf{T}(\boldsymbol{\xi}) \cdot \boldsymbol{\pi}^{-1}(\mathbf{q}, d_{\mathbf{q}})) \in \Omega_k \} \end{aligned} \quad (3.19)$$

where  $\mathbf{w}_n$ , with  $m = 2, \dots, N_l - 1$  referring to the intermediate points defined homogeneously along the line segments.

Then, the intensity residual for a line segment  $\delta I_l$  is defined as the photometric difference between pixels of the same 3D line segment point, which is:

$$\delta I_l(\boldsymbol{\xi}, \mathbf{l}) = \frac{1}{N_l} \sum_{n=0}^{N_l} \left| I_k \left( \pi(\mathbf{T}(\boldsymbol{\xi}) \cdot \mathbf{w}_n) \right) - I_{k-1}(\mathbf{w}_n) \right| \quad (3.20)$$

where in the case of  $n = 0$  and  $n = N_l$ , the point  $\mathbf{w}_n$  refers to the endpoints  $\mathbf{p}$  and  $\mathbf{q}$  respectively. Then, we estimate the optimal pose increment  $\boldsymbol{\xi}_{k-1,k}^*$  that minimizes the photometric error of all patches, for both point and line segment features:

$$\boldsymbol{\xi}_{k-1,k}^* = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \left\{ \sum_{i \in \mathcal{P}} \|\delta I_p(\boldsymbol{\xi}, \mathbf{x}_i)\|^2 + \sum_{j \in \mathcal{L}} \|\delta I_l(\boldsymbol{\xi}, \mathbf{l}_j)\|^2 \right\}. \quad (3.21)$$

Similarly to [53], we employ inverse compositional formulation proposed in [17], for speeding up the minimization process. In this case, we seek for the linearized Jacobian of the line segment residuals, which can be expressed as

the summatory of the individual point Jacobians for each intermediate point  $\mathbf{w}_n$  sampled:

$$\left. \frac{\partial \delta I_l(\boldsymbol{\xi}, \mathbf{l}_j)}{\partial \boldsymbol{\xi}} \right|_{\boldsymbol{\xi}=\mathbf{0}} = \frac{1}{N_l} \sum_{m=0}^{N_l} \left. \frac{\partial \delta I_p(\boldsymbol{\xi}, \mathbf{w}_n)}{\partial \boldsymbol{\xi}} \right|_{\boldsymbol{\xi}=\mathbf{0}} \quad (3.22)$$

whose expression can be obtained of [53]. Then, we estimate the optimal pose by robust Gauss-Newton minimization of the above-mentioned cost function in (3.21). Notice that this formulation allows for the fast tracking of line segments as depicted in Figure 3.10, which is an open problem due to the high computational burden employed with traditional feature-based approaches [62].

### 3.E.3.2 Individual Feature Alignment

Similarly to [53], we individually refine the 2D positions of each feature by minimizing the photometric error between the patch in the current image, and the projection of all the 3D observations of this feature, which can be solved by employing Lucas-Kanade algorithm [17]. In the case of line segments, we only need to refine the position of the 2D endpoints (see Figure 3.13), which defines the line equation employed in the estimation of the projection errors:

$$\mathbf{w}'_j = \underset{\mathbf{w}'_j}{\operatorname{argmin}} \left\| I_k(\mathbf{w}'_j) - I_r(\mathbf{A}_j \cdot \mathbf{w}_j) \right\|^2, \forall j \quad (3.23)$$

where  $\mathbf{w}'_j$  is the 2D estimation of the position of the feature in the current frame ( $\mathbf{w}'_j$  stands equally for both endpoints), and  $\mathbf{w}_j$  is the position of the feature in the reference frame  $r$ . This is a bold assumption in the case of line segments, since their endpoints are considerably less descriptive than key-points. For dealing with this, we also perform a robust optimization of (3.23), and then we relax this assumption by refining the 3D position of the endpoints. Notice that it is necessary to employ an affine warping  $\mathbf{A}_j$  in this step, since the closest key frame for which we project the feature is usually farther, and the size of the patch is bigger than in the previous step.

### 3.E.3.3 Pose and Structure Refinement

After optimizing individually the position of each feature in the image by skipping the epipolar constraints, the camera pose obtained in (3.21) must be further refined by minimizing the reprojection errors between the 3D features and the corresponding 2D feature positions in the image (see Figure 3.14). For that, we consider reprojection errors between the 3D features and the camera pose  $T_{k,w}$ , both in world's coordinate frame, since it considerably reduces the

drift of the estimated trajectory. The cost function when employing both type of features is:

$$\begin{aligned} \xi_{k,w}^* = \operatorname{argmin}_{\xi} \Big\{ & \sum_{i \in \mathcal{P}} \|r_p(\mathbf{T}_{k,w}, \mathbf{X}_{i,k})\|^2 \\ & + \sum_{j \in \mathcal{L}} \|r_l(\mathbf{T}_{k,w}, \mathbf{P}_{j,k}, \mathbf{Q}_{j,k}, \mathbf{l}_j)\|^2 \Big\} \end{aligned} \quad (3.24)$$

where  $r_p$  stands for the projection errors in the case of point features, and  $r_l$  is the projection error of line segments:

$$r_l(T_{k,w}, \mathbf{P}_{j,k}, \mathbf{Q}_{j,k}, \mathbf{l}_j) = \begin{bmatrix} \mathbf{l}_j \cdot \boldsymbol{\pi}(\mathbf{T}_{k,w} \cdot \mathbf{P}_{j,k}) \\ \mathbf{l}_j \cdot \boldsymbol{\pi}(\mathbf{T}_{k,w} \cdot \mathbf{Q}_{j,k}) \end{bmatrix}. \quad (3.25)$$

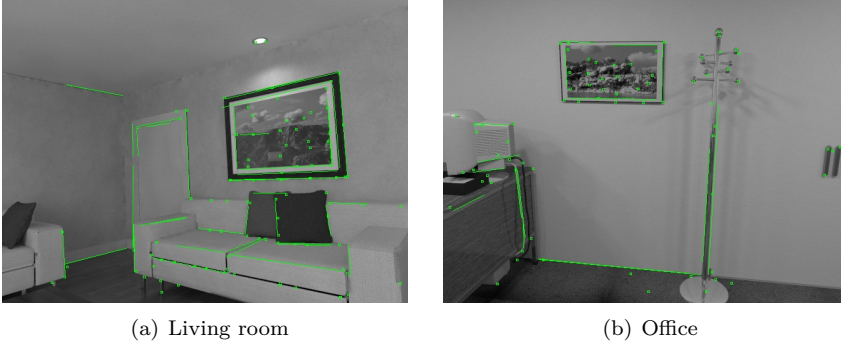
This is solved iteratively with Gauss-Newton, for which we need to include (3.24) in a robustified framework, for which we employ the Cauchy loss function:

$$\rho(s) = \log(1 + s). \quad (3.26)$$

The optimization consist of three steps: i) we first estimate the camera motion with all the samples ii) we filter out the outliers, which are considered as those features whose residual error lies above two times the robust standard deviation of those errors iii) we fastly refine the camera pose by optimizing with the inliers subset. Finally, we refine the position of the 3D point and line segment features through minimization of the reprojection errors.

### 3.E.4 Mapping

The map thread recursively estimates the 3D position of the image features for which their depth is still unknown. For that, authors of [53] implement a depth filter based in a Bayesian framework, for which they model the depth of the feature with a Gaussian + Uniform mixture model distribution [146]. In the case of line segments we need to estimate the 3D position of the endpoints, since they are employed for both describe the feature and estimating the reprojection errors. However, the endpoints of line segments obtained through detectors such as LSD [147] are not repetitive, which is a limitation to employ them in monocular visual odometry. On the other hand, one of the advantages of the fast tracking employed here is that we explicitly seek for the exact same line segment in the successive frames, so that we continuously track the position of the endpoints. This allows for the introduction of the endpoints in a similar Bayesian framework, where the distribution of both endpoints is estimated when inserting new observations. As a result, we obtain meaningful maps which can be used to extract useful information about geometry of the scene.



**Figure 3.15:** Sparse features tracked by PL-SVO in two frames extracted from the ICL-NUIM dataset [68], where we can observe the importance of introducing line segments in such low-textured scenarios.

### 3.E.5 Experimental Validation

In this section, we illustrate the benefits of including line segments in motion estimation, specially when working in low-textured environments. For that, we estimate the trajectory of a monocular camera in several sequences, from both synthetic and real datasets. All experiments have been conducted on an Intel Core i5-6600 CPU @ 3.30GHz without GPU parallelization.

#### 3.E.5.1 Evaluation in ICL-NUIM Dataset [68]

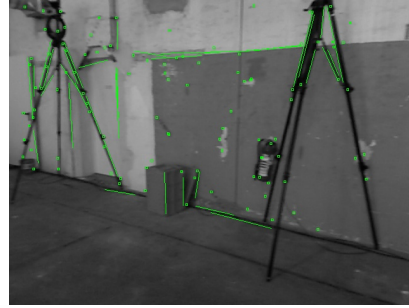
First, we test our algorithm in the Imperial College London and National University of Ireland Maynooth (ICL-NUIM) dataset [68]. This dataset consist of two different synthetic environments, one in an office and the other one in a living room, for which several sequences can be generated and rendered (see Figure 3.15). Table 3.6 compares the performance of the proposed algorithm against SVO [53] for the sequences *lrkt-2* and *ofkt-3*. For the sake of fairness, we have employed our current implementation of PL-SVO without introducing line segments to the framework as baseline of comparison. Results show a superior performance of PL-SVO in the first sequence *lrkt-2*, which is capable of estimating the camera motion along the whole trajectory (the rest of the sequence is employed for initialization), while the point-based approach only tracks the 34% of the sequence. In the second sequence, *ofkt-3*, SVO shows a slight superiority in terms of accuracy. However, it is worth noticing that it is only capable of tracking a 57%, and hence, it is not affected by higher errors introduced in the difficult parts of the sequence.

**Table 3.5:** Comparison against SVO [53] in the ICL-NUIM dataset by measuring relative pose errors (RPE) per second.

		% Sequence	RMSE (m/s)	Median (m/s)	RMSE (deg/s)	Median (deg/s)
<i>lrkt-2</i>	SVO	34.32	0.0085	0.0078	<b>0.0615</b>	<b>0.0491</b>
	PL-SVO	<b>91.93</b>	<b>0.0076</b>	<b>0.0069</b>	0.1023	0.0502
<i>ofkt-3</i>	SVO	57.82	<b>0.0053</b>	<b>0.0041</b>	0.1011	0.0547
	PL-SVO	<b>96.85</b>	0.0059	0.0053	<b>0.0997</b>	<b>0.0532</b>



(a) Textured scene



(b) Low-textured scene

**Figure 3.16:** Sparse features tracked by PL-SVO in two different frames of the TUM dataset [68].

### 3.E.5.2 Evaluation in TUM Dataset [140]

We also evaluate the performance of both SVO and PL-SVO approaches in the TUM Dataset [140], which consist of several sequences recorded with an RGB-D camera in different environments, as depicted in Figure 3.16. Table

**Table 3.6:** Comparison against SVO [53] in the TUM dataset by measuring the RPE per second.

		% Sequence	RMSE (m/s)	Median (m/s)	RMSE (deg/s)	Median (deg/s)
<i>fr1-floor</i>	SVO	54.47	0.4112	<b>0.0528</b>	21.0040	2.1970
	PL-SVO	<b>77.11</b>	<b>0.0806</b>	0.0742	<b>2.2658</b>	<b>1.1294</b>
<i>fr1-xyz</i>	SVO	95.10	0.1780	0.1251	11.8003	8.8365
	PL-SVO	95.10	<b>0.1089</b>	<b>0.0873</b>	<b>7.6256</b>	<b>5.9863</b>
<i>fr2-desk</i>	SVO	96.23	0.0908	0.0828	0.9912	0.7675
	PL-SVO	96.23	<b>0.0693</b>	<b>0.0644</b>	<b>0.9040</b>	<b>0.7275</b>
<i>fr2-rpy</i>	SVO	98.39	<b>0.0155</b>	<b>0.0100</b>	<b>0.6424</b>	<b>0.5037</b>
	PL-SVO	98.39	0.0157	0.0107	0.6501	0.5200
<i>fr2-xyz</i>	SVO	98.56	0.0213	0.0183	0.6462	0.5449
	PL-SVO	98.56	<b>0.0209</b>	<b>0.0178</b>	<b>0.6337</b>	<b>0.4845</b>
<i>fr3-longoffice</i>	SVO	95.35	0.1794	0.1793	<b>1.1132</b>	0.6138
	PL-SVO	95.35	<b>0.1660</b>	<b>0.1637</b>	1.3118	<b>0.5161</b>

3.5 contains the results for the considered sequences from the TUM dataset. In general, we observe the superior performance of PL-SVO in most sequences, hence confirming its robust behavior in multiple environments. However, the accuracy of motion estimation considerably decreases in this dataset, where monocular techniques are severely affected by motion blur and other negative effects resulting from the rolling shutter in RGB-D sensors.

### 3.E.5.3 Processing Time

Finally, we analyze the impact of introducing line segments to the framework in the processing time. Table 3.7 shows the average times employed in the different stages of the algorithms. As one may first think, the computational cost necessary for performing both sparse image and feature alignment increases considerably when including line segments, where the runtime of each stage is augmented in 4 ms. However, our algorithm still performs in real-time with frequencies of almost 60 Hz, depending on the type of scene.

**Table 3.7:** Mean average times in each stage of the algorithm for both SVO and PL-SVO algorithms.

	SVO [53]	PL-SVO
Pyramid creation	0.26 ms	0.26 ms
Sparse Image Alignment	2.60 ms	6.58 ms
Feature Alignment	4.13 ms	8.61 ms
Pose and Structure Refinement	0.35 ms	0.76 ms
Total Motion Estimation:	<b>8.60 ms</b>	<b>17.83 ms</b>

### 3.E.6 Conclusions

In this paper we have proposed a novel approach to monocular odometry by extending the SVO algorithm proposed by Forster et al. in [53] to the case of line segments. Hence, we obtain a more robust system capable of dealing with untextured environments, where performance of point-based approaches usually deteriorates due to the difficulties in finding a well-distributed set of points. The semi-dense approach allows for the fast tracking of line segments, thus eliminating the necessity of detecting and matching whenever a new frame is introduced, which is one of the main limitations of employing this type of features. We validate the claimed features in a series of experiments in both synthetic and real datasets, confirming the robust behavior of this proposal.

---

## 3.F PL-SLAM: a Stereo SLAM System through the Combination of Points and Line Segments

---

Ruben Gomez-Ojeda, Francisco-Angel Moreno, David Zuñiga-Noël,  
Davide Scaramuzza, and Javier Gonzalez-Jimenez

*Transactions on Robotics (T-RO), 2019.*

©IEEE (Revised layout)

# PL-SLAM: a Stereo SLAM System through the Combination of Points and Line Segments

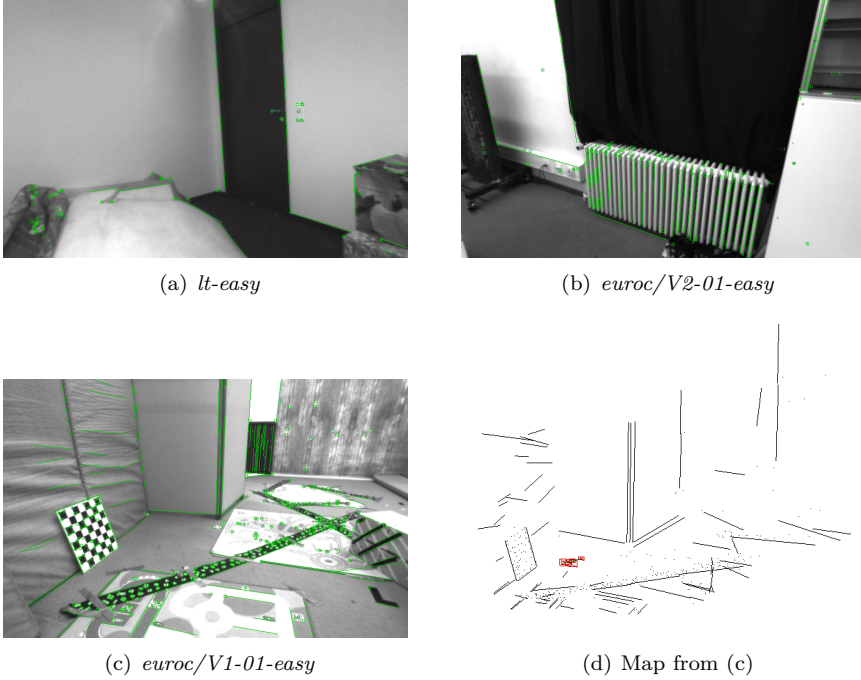
*Ruben Gomez-Ojeda, Francisco-Angel Moreno, David Zuñiga-Noël,  
Davide Scaramuzza, and Javier Gonzalez-Jimenez*

## Abstract

Traditional approaches to stereo visual SLAM rely on point features to estimate the camera trajectory and build a map of the environment. In low-textured environments, though, it is often difficult to find a sufficient number of reliable point features and, as a consequence, the performance of such algorithms degrades. This paper proposes PL-SLAM, a stereo visual SLAM system that combines both points and line segments to work robustly in a wider variety of scenarios, particularly in those where point features are scarce or not well-distributed in the image. PL-SLAM leverages both points and line segments at all the instances of the process: visual odometry, keyframe selection, bundle adjustment, etc. We contribute also with a loop closure procedure through a novel bag-of-words approach that exploits the combined descriptive power of the two kinds of features. Additionally, the resulting map is richer and more diverse in 3D elements, which can be exploited to infer valuable, high-level scene structures like planes, empty spaces, ground plane, etc. (not addressed in this work). Our proposal has been tested with several popular datasets (such as EuRoC or KITTI), and is compared to state of the art methods like ORB-SLAM2, revealing a more robust performance in most of the experiments, while still running in real-time. An open source version of the PL-SLAM C++ code has been released for the benefit of the community.

### 3.F.1 Introduction

In recent years, visual Simultaneous Localization And Mapping (SLAM) has been firmly progressing towards the degree of reliability required for fully autonomous vehicles: mobile robots, self-driving cars or Unmanned Aerial Vehicles (UAVs). In a nutshell, the SLAM problem consists of the estimation of the vehicle trajectory given as a set of poses (position and orientation), while simultaneously building a map of the environment. Apart from self-localization, a map becomes useful for obstacle avoidance, object recognition, task planning, etc. [44].



**Figure 3.17:** Low-textured environments are challenging for *feature-based* SLAM systems based on traditional keypoints. In contrast, line segments are usually common in human-made environments, and apart from an improved camera localization, the built maps are richer as they are populated with more meaningful elements (3D line-segments).

As a first-level classification, SLAM systems can be divided into *topological* (e.g. [35, 107–109]) and *metric* approaches. In this paper, we focus on the latter, which take into account the *geometric* information of the environment and build a physically meaningful map of it [89, 113]. These approaches can be further categorized into *direct* and *feature-based* systems.

*Direct* methods estimate the camera motion by minimizing the photometric errors between consecutive frames under the assumption of constant brightness along the local parts of the sequences (examples of this approach can be found elsewhere [50, 54, 118]). While this group of techniques has the advantage of working directly with the input images regardless of any intermediate representation, they are very sensitive to brightness changes (this phenomena was addressed in [49]) and constrained to narrow baseline motions. In contrast, *feature-based* methods employ an indirect representation of the images,

typically in the form of point features, that are tracked along the successive frames and then employed to recover the pose by minimizing the projection errors [115, 128].

It is noticeable that the performance of any of the above-mentioned approaches usually decreases in low-textured environments in which it is typically difficult to find a large set of keypoint features. The effect in such cases is an accuracy impoverishment and, occasionally, the complete failure of the system. Many of such low-textured environments, however, contain planar elements that are rich in linear shapes, so it would be possible to extract line segments from them. We claim that these two types of features (keypoints and segments) complement each other and its combination leads to a more versatile, robust and stable SLAM system. Furthermore, the resulting maps comprising both 3D points and line segments provide more structural information from the environment than point-only maps, as can be seen in the example shown in Figure 3.17(d). Thus, applications that perform high-level tasks such as place recognition, semantic mapping or task planning, among others, can significantly benefit from the richer information that can be inferred from them.

These benefits, though, come at the expense of a higher computational burden in both detecting and matching line-segments in images [18], and also in dealing effectively with segment-specific problems like partial occlusions, line disconnection, etc., which complicate feature tracking and matching as well as the residual computation for the map and pose optimization. Such hurdles are the reason why the number of solutions that have been proposed in the literature to SLAM or Structure from Motion (SfM) with line features (e.g. [22, 72, 106, 136, 153]) is so limited. Besides, the few solutions we have found only perform robustly in highly structured environments while showing unreliable results when applied to more realistic ones such as those recorded in the EuRoC or KITTI datasets. In this work, we address the segment-specific tracking and matching issues by discarding outliers through the comparison of the length and the orientation of the line features, while, for the residual computation, we represent segments in the map by their endpoints coordinates. Thus, the residuals between the observed segments and their corresponding lines in the map are computed by the distance between the projections of those endpoints on the image plane and the infinite lines associated to the observed ones. This way, we are able to build a consistent cost function that seamlessly encompasses both point and line features.

These two kinds of features are also employed to robustly detect loop closures during camera navigation, following a new bag-of-words approach that combines the advantages of using each of them to perform place recognition. In summary, we propose a novel and versatile stereo visual SLAM system, coined PL-SLAM, which builds upon our previous Visual Odometry (VO) approach presented in [62], and combines both point and line segment features to per-

form real-time camera localization and mapping. The main contributions of this work are:

- The first open source stereo SLAM system that employs point and line segment features in real time, hence being capable of operating robustly in low-textured environments where traditional point-only approaches tend to fail. Because of the consideration of both kinds of features, our proposal also produces rich geometrical maps.
- A new implementation of the bundle adjustment (BA) process that seamlessly accounts for both kinds of features while refining the poses of the keyframes.
- An extension of the bag-of-words approach presented in [57] that takes into account the description of both points and line segments to improve the loop-closure process.

A set of illustrative videos showing the performance of proposed system and an open source version of the developed C++ PL-SLAM library are publicly available at <http://mapir.uma.es> and <https://github.com/rubengooj/pl-slam>.

### 3.F.2 Related Work

Feature-based SLAM is traditionally addressed by tracking keypoints along successive frames and then minimizing some error function (typically based on re-projection errors) to simultaneously estimate the poses and the map [25]. Among the most successful proposals, we can highlight FastSLAM [110], PTAM [88] [87], SVO [53] [54], and, more recently, ORB-SLAM [115], which relies on a fast and continuous tracking of ORB features [126], and a local bundle adjustment step with the continuous observations of the point features. All these approaches, though, tend to fail or reduce their accuracy in low-textured scenarios where the lack of repeatable and reliable features usually hinders the feature tracking process. In the following, we review the state of the art of visual SLAM systems based on alternative image features to keypoints: i.e. edgelets, lines, or line segments.

One of the remarkable approaches that employs *line* features is the one in [134], where the authors proposed an algorithm to integrate them into a monocular Extended Kalman Filter SLAM system (EKF-SLAM). In the cited paper, the line detection relies on an hypothesize-and-test method that connects several nearby keypoints to achieve real-time performance. Other works employ *edge* landmarks as features in monocular SLAM, as the one reported in [47], which does not only include the information of the local planar patch as in the case of keypoints, but also considers local edge segments, hence introducing new valuable information as the orientation of the so-called *edgelets*. In that work they derive suitable models for those kinds of features and use them

within a particle-filter SLAM system, achieving nearly real-time performance. More recently, authors in [54] also introduced edgelets in combination with intensity corners in order to improve robustness in environments with little or high-frequency texture.

A different approach, known as *model-based*, incorporates prior information about the orientation of the landmarks derived from line segments. Particularly, the method in [152] presents a monocular 2D SLAM system that employs vertical and horizontal lines on the floor as features for both motion and map estimation. For that, they propose two different parameterizations for the vertical and the horizontal lines: vertical lines are represented as 2D points on the floor plane (placed at the intersection point between the line and such plane), while horizontal lines are represented by their two end-points placed on the floor. Finally, the proposed model is incorporated into an EKF-SLAM system. Another model-based approach is reported in [157], where the authors introduce structural lines in an extension of a standard EKF-SLAM system. The dominant directions of the lines are estimated by computing their vanishing points under the assumption of a Manhattan world [34]. All these model-based approaches, though, are limited to very structured scenarios and/or planar motions, as they rely solely on line features.

The works in [135, 136] address a generic approach that compares the impact of eight different landmark parametrization for monocular EKF-SLAM, including the use of point and line features. Nevertheless, such systems are only validated through analytic and statistical tools that assumed already known data association and that, unlike our proposal, do not implement a complete front-end that detect and track the line segments. Another technique for building a 3D line-based SLAM system has been proposed in the recent work [151]. For that, the authors employ two different representations for the line segments: the Plücker line coordinates for the initialization and 3D projections, and an orthonormal representation for the back-end optimization. Unfortunately, neither the source code is available nor the employed dataset contains any ground-truth, therefore it has not been possible to compare with our proposal.

Recently, line segment features have also been employed for monocular pose estimation in combination with points, due to the bad-conditioned nature of this problem. For that, in [61] the authors extended the semi-direct approach in [53] with line segments. Thanks to this pipeline, line segments can be propagated efficiently throughout the image sequence, while refining the position of the end-points under the assumptions of high frame rate and very narrow-baseline.

Finally, by the time of the first submission of this paper, a work with the same name (PL-SLAM, [124]) was published extending the monocular algorithm ORB-SLAM to the case of including line segment features computed through the LSD detector [147]. Apart from being a monocular system (unlike our stereo approach), their proposal deals with line tracking and matching in an

essentially different way: they propagate the line segments by their endpoints and then perform descriptor-based tracking, which decreases the time performance of ORB-SLAM. Besides this computational drawback, when working with features detected with the LSD detector, the variance of the endpoints becomes quite pronounced, specially in challenging illumination conditions or very low-textured scenes, making more difficult wide-baseline tracking and matching between line features in non-consecutive frames. Our PL-SLAM approach, in contrast, does not make any assumption regarding the position of the lines endpoints so that our tracking front-end allows to handle partially occluded line segments, endpoints variance, etc., for both the stereo and frame-to-frame tracking, hence becoming a more robust approach to point-and-line SLAM.

### 3.F.3 PL-SLAM Overview

The general structure of the PL-SLAM system proposed here is depicted in Figure 3.18, while its main modules will be presented in the following sections. This structure is strongly based on the scheme first proposed by ORB-SLAM [115] and also implements three different threads: *visual odometry*, *local mapping*, and *loop closure*. This efficient distribution allows for a continuous tracking of the VO module while the local mapping and the loop closure ones are processed in the background only when a new keyframe is inserted (other approaches that exploits parallel threads can be found elsewhere [88, 118]). As will be further described, our proposal also takes some of the ORB-SLAM ideas as basis for developing our point-and-line SLAM system.

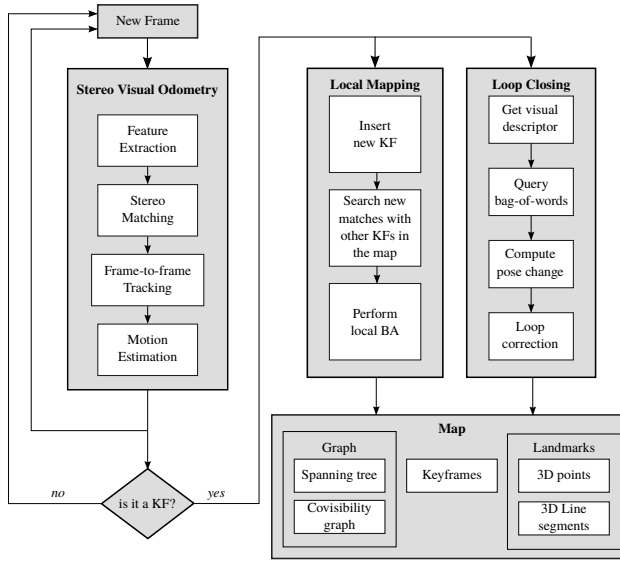
#### Map

The map consists of i) a set of keyframes (KFs), ii) the detected 3D landmarks (both keypoints and line segments), iii) a covisibility graph and iv) a spanning tree.

The keyframes contain the observed stereo features and their descriptors, a visual descriptor of the corresponding left image computed through a visual vocabulary, as explained later in Section 3.F.6, and the information of the 3D camera pose.

Regarding the landmarks, we store the list of observations and the most representative descriptor for each detected landmark. Besides, specifically for points, we also keep its estimated 3D position while, for the line segments, we keep both their direction and the estimated 3D coordinates of their endpoints.

Finally, the covisibility information is modeled by a graph (as in [139]), where each node represents a KF, and the edges between KFs are created only if they share a minimum number of landmarks (which in this work is set to 20), allowing for real-time bundle adjustment of the local map. Please, refer to Figure 3.19 for an example of a covisibility graph.



**Figure 3.18:** Scheme of the stereo PL-SLAM system.

In order to perform a faster loop closure optimization, we also form the so-called *essential graph*, which is less dense than the covisibility graph as an edge between two KFs is created only when they share more than 100 landmark observations. Finally, the map also contains a *spanning tree*, which stands for the minimum connected representation of a graph that includes all the KFs. Both the essential graph and the use of a spanning tree for loop closure are ideas originally proposed in [115].

### Feature Tracking

We perform feature tracking through the stereo visual odometry algorithm from our previous work [62]. In a nutshell, we track image features (points and segments) from a sequence of stereo frames and compute their 3D position and their associated uncertainty represented by covariance matrices. The 3D landmarks are then projected to the new camera pose, and the projection errors are minimized in order to obtain both the camera pose increment and the covariance associated to such estimation. This process is repeated every new frame, performing simply frame to frame VO, until a new KF is inserted into the map. Further discussion about the feature tracking procedure will be formally addressed in Section 3.F.4.

Once a KF is inserted into the map, two procedures are run in parallel: local mapping and loop closure detection.

## Local Mapping

The local mapping procedure looks for new feature correspondences between the new KF, the last one and those connected to the last one in the covisibility graph. This way, we build the so-called *local map* of the current KF, which includes all the KFs that share at least 20 landmark observations with the current one as well as all the landmarks observed by them. Finally, an optimization of all the elements within the local map (i.e. KF poses and landmarks positions) is performed. A detailed description of this procedure will be presented in Section 3.F.5.

## Loop Closure

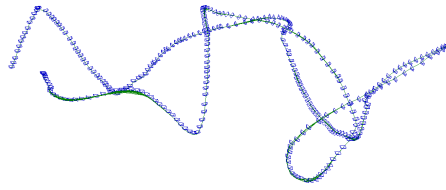
In parallel to local mapping, loop closure detection is carried out by extracting a visual descriptor for each image, based on a bag-of-words approach, as will be described in Section 3.F.6. All the visual descriptors of the captured keyframes during camera motion are stored in a database, which is later employed to find similar frames to the current one. The best match will be considered a loop closure candidate only if the local sequence surrounding this KF is also similar. Finally, the relative  $SE(3)$  transformation between the current KF and the loop closure candidate is estimated so that, if a proper estimation is found, all the KFs poses involved in the loop are corrected through a pose-graph optimization (PGO) process.

It is important to remark that the stereo visual odometry system runs continuously at every frame while both the local mapping and loop closure detection procedures are launched in the background (in separated threads) only when a new KF is inserted, thus allowing our system to reach real-time performance. In the event of a new keyframe being inserted in the system while the local mapping thread is still being processed, the keyframe is temporary stored until the map is updated and then a new local mapping process is launched.

It is worth mentioning that, as declared before, these mapping and loop closure pipelines are identical to the ones presented in ORB-SLAM, being aimed to reduce (along with the incorporation of recent sparse algebra techniques) the high computational burden that general BA involves. Within the BA framework, our proposal belongs to the so-called *relative* techniques (e.g. [113, 130, 131]), which have gained great popularity in the last years as an alternative to the more costly *global* approaches (e.g. [80, 88]).

## 3.F.4 Feature Tracking

This section reviews the most important aspects of our previous work [62], which deals with the visual odometry estimation between consecutive frames, and also with the KF decision policy. Briefly, both points and line segments are tracked along a sequence of stereo frames (see Figure 3.17), and, then,



**Figure 3.19:** Covisibility graph in the sequence *lt-first* for which we have represented the edges connecting the keyframes with green lines.

both the 3D motion of the camera and its uncertainty are computed through the minimization of the projection errors. Note that this process only performs the optimization of the camera poses and not the 3D position of the tracked features, whose coordinates in the map are refined during the local bundle adjustment procedure explained in the next section.

Since the stereo cameras employed in the experiments are pre-calibrated, the initialization of these features is performed in a single stereo shot by straightforwardly employing their extrinsic parameters to determine the 3D position of both the observed keypoints and lines.

### Point Features

In this work we use the well-known ORB method [126] due to its great performance for keypoint detection, and the binary nature of the descriptor it provides, which allows for a fast, efficient keypoint matching. In order to reduce the number of outliers, we only consider measurements which fulfill that the best match in the left image corresponds to the best match in the right one, i.e. they are mutual best matches. Finally, we also filter out those matches whose distance in the descriptor space with the second best match is less than twice the distance with the best match, to ensure that the correspondences are meaningful enough.

### Line Segment Features

The Line Segment Detector (LSD) method [147] has been employed to extract line segments, providing high precision and repeatability. For stereo matching and frame-to-frame tracking we augment line segments with a binary descriptor provided by the Line Band Descriptor (LBD) method [154], which allows us to find correspondences between lines based on their local appearance. Similarly to the case of points, we check that both candidate features are mutual best matches, and also that the feature is meaningful enough. Finally, we take advantage of the useful geometrical information that line segments provide in order to filter out those line matches with different orientations and lengths, and those with a high difference on the disparities of the endpoints. Notice

that this filter helps the system to retain a larger amount of structural lines, which allows the formation of more consistent maps based on points and lines (see Figure 3.17(d)).

### Motion Estimation

Once we have established the correspondences between two stereo frames, we back-project both the keypoints and the line segments from the first frame to the next one. Then, we iteratively estimate the camera ego-motion through a robust Gauss-Newton minimization of the line and keypoint projection errors. In order to deal with outliers, we employ a Pseudo-Huber loss function and perform a two-step minimization, as proposed in [112]. Finally, we obtain the incremental motion estimation between the two consecutive frames, which can be modelled by the following normal distribution:

$$\xi_{t,t+1} \sim \mathcal{N}(\xi_{t,t+1}^*, \Sigma_{\xi_{t,t+1}}^*) \quad (3.27)$$

where  $\xi_{t,t+1}^* \in \mathfrak{se}(3)$  is the 6D vector of the camera motion between the frames  $t$  and  $t+1$ , and  $\Sigma_{\xi_{t,t+1}}^*$  stands for the covariance of the estimated motion, approximated by the inverse of the Hessian of the cost function in the last iteration.

### Keyframe Selection

For deciding when a new KF is inserted in the map, we have followed the approach in [83] which employs the uncertainty of the relative motion estimation. Thus, following equation (3.27), we transform the uncertainty from the covariance matrix into a scalar, named *entropy*, through the following expression:

$$h(\xi) = 3(1 + \log(2\pi)) + 0.5 \log(|\Sigma_{\xi}|) \quad (3.28)$$

Then, for a given KF  $i$  we check the ratio between the entropy from the motion estimation between the previous KF  $i$  and the current one  $i+u$  and that between the previous KF  $i$  and its first consecutive frame  $i+1$ , i.e.:

$$\alpha = \frac{h(\xi_{i,i+u})}{h(\xi_{i,i+1})} \quad (3.29)$$

If the value of  $\alpha$  lies below some pre-established threshold, which in our experiments has been set to 0.9, then the frame  $i+u$  is inserted to the system as a new KF. Notice that to compute the expression in Equation (3.28), we need the uncertainty of the pose increment between non-consecutive frames. Since Equation (3.27) only estimates the incremental motion between consecutive frames, a series of such estimations are composed through first order Gaussian propagation techniques to obtain the covariance between two non-consecutive KFs.

### 3.F.5 Local Mapping

This section describes the behavior of the system when a new KF is inserted, which essentially consists in performing the bundle adjustment of the so-called *local map* i.e.: those KFs connected with the current one by the covisibility graph and the landmarks observed by those local KFs.

#### Keyframe Insertion

Every time the visual odometry thread selects a KF, we insert it into the SLAM system and optimize the local map. First, we refine the estimation of the relative pose change between the current and the previous KFs, since the one provided by the VO is estimated by composing the relative motions between the intermediate frames. For that, we perform data association between the KFs, taking into account the geometrical restrictions described in Section 3.F.4 and obtaining a consistent set of common features observed in them. Then, we perform a similar optimization than the one presented in Section 3.F.4, for which we employ the pose provided by the VO thread as the initial estimation for a Gauss-Newton minimization. Once we have computed the relative pose change between the KFs, we insert the current one into the system, including:

1. An index for the keyframe.
2. The information of its 3D pose, which comprises an absolute pose and the relative pose from the previous KF, along with their associated uncertainties.
3. The new 3D landmarks, which are initialized by storing both their 2D image coordinates and their descriptors. The new observations of the already existing landmarks are also added to the map.

Finally, we also look for new correspondences between the unmatched feature observations from the current frame, and the landmarks in the local map.

#### Local Bundle Adjustment

After inserting the KF, the next step is to perform a bundle adjustment of the local map. As stated before, this map is formed by all the KFs connected with the current one in the covisibility graph (i.e. those that share at least 20 landmarks) and also all the landmarks observed by the local KFs. For that, let us define the vector  $\psi$  that contains the variables to be optimized, which are the  $\mathfrak{sc}(3)$  pose of each KF  $\xi_{iw}$ , the 3D position of each point  $\mathbf{X}_{wj}$ , and also the 3D positions of the endpoints for each line segment:  $\{\mathbf{P}_{wk}, \mathbf{Q}_{wk}\}$ . Then,

we minimize the projection errors between the observations and the landmarks projected to the frames where they were observed:

$$\psi^* = \underset{\psi}{\operatorname{argmin}} \sum_{i \in \mathcal{K}_l} \left[ \sum_{j \in \mathcal{P}_l} \mathbf{e}_{ij}^\top \Sigma_{\mathbf{e}_{ij}}^{-1} \mathbf{e}_{ij} + \sum_{k \in \mathcal{L}_l} \mathbf{e}_{ik}^\top \Sigma_{\mathbf{e}_{ik}}^{-1} \mathbf{e}_{ik} \right] \quad (3.30)$$

where  $\mathcal{K}_l$ ,  $\mathcal{P}_l$  and  $\mathcal{L}_l$  refer to the groups of local KFs, points, and line segments, respectively.

In this expression, the projection error  $\mathbf{e}_{ij}$  stands for the 2D distance between the observation of the  $j$ -th map point in the  $i$ -th KF, and can be expressed as:

$$\mathbf{e}_{ij} = \mathbf{x}_{ij} - \pi(\xi_{iw}, \mathbf{X}_{wj}) \quad (3.31)$$

where the function  $\pi: \mathfrak{se}(3) \times \mathbb{R}^3 \mapsto \mathbb{R}^2$  first places the  $j$ -th 3D point  $\mathbf{X}_{wj}$  (in world coordinates) into the local reference system of the  $i$ -th KF, i.e.  $\mathbf{X}_{ij}$ , and then projects this point to the image. The use of line segments is slightly different, since we cannot simply compare the position of the endpoints as they might be displaced along the line or occluded from one frame to the next one. For that, we take as error function the distances between the projected endpoints of the 3D line segment and its corresponding infinite line in the image plane. In this case, the error  $\mathbf{e}_{ik}$  between the  $k$ -th line observed in the  $i$ -th frame, is given by:

$$\mathbf{e}_{ik} = \begin{bmatrix} \mathbf{l}_{ik} \cdot \pi(\xi_{iw}, \mathbf{P}_{wk}) \\ \mathbf{l}_{ik} \cdot \pi(\xi_{iw}, \mathbf{Q}_{wk}) \end{bmatrix} \quad (3.32)$$

where  $\mathbf{P}_{wk}$  and  $\mathbf{Q}_{wk}$  refer to the 3D endpoints of the line segments in the world coordinate system and  $\mathbf{l}_{ik}$  is the equation of the infinite line that corresponds to the  $k$ -th line segment in the  $i$ -th KF, which can be obtained with the cross product between the 2D endpoints of the line segments in homogeneous coordinates, i.e.:  $\mathbf{l}_{ik} = \mathbf{p}_{ik} \times \mathbf{q}_{ik}$ .

The problem in (3.30) can be iteratively solved by following the Levenberg-Marquardt optimization approach, for which we need to estimate both the Jacobian and the Hessian matrices:

$$\Delta\psi = [\mathbf{H} + \lambda \operatorname{diag}(\mathbf{H})]^{-1} \mathbf{J}^\top \mathbf{W} \mathbf{e} \quad (3.33)$$

where the error vector  $\mathbf{e}$  contains all the projection errors  $\mathbf{e}_{ij}$  and  $\mathbf{e}_{ik}$ . This equation, along with the following update step:

$$\psi' = \psi \boxplus \Delta\psi \quad (3.34)$$

can be applied recursively until convergence, resulting in the optimal  $\psi$ , from which we can update the position of the local KFs and landmarks. Notice that the update equation cannot be directly applied to the whole vector, given the different nature of the variables in  $\psi$ .

$$\mathbf{H}_{i,jk} = \begin{bmatrix} \vdots & \frac{\partial \mathbf{e}_{ij}}{\partial \boldsymbol{\xi}_{iw}}^\top \boldsymbol{\Sigma}_{\mathbf{e}_{ij}}^{-1} \frac{\partial \mathbf{e}_{ij}}{\partial \boldsymbol{\xi}_{iw}} + \frac{\partial \mathbf{e}_{ik}}{\partial \boldsymbol{\xi}_{iw}}^\top \boldsymbol{\Sigma}_{\mathbf{e}_{ik}}^{-1} \frac{\partial \mathbf{e}_{ik}}{\partial \boldsymbol{\xi}_{iw}} & \vdots & \vdots & \frac{\partial \mathbf{e}_{ij}}{\partial \boldsymbol{\xi}_{iw}}^\top \boldsymbol{\Sigma}_{\mathbf{e}_{ij}}^{-1} \frac{\partial \mathbf{e}_{ij}}{\partial \mathbf{X}_{wj}} & \vdots & \vdots & \frac{\partial \mathbf{e}_{ik}}{\partial \boldsymbol{\xi}_{iw}}^\top \boldsymbol{\Sigma}_{\mathbf{e}_{ik}}^{-1} \frac{\partial \mathbf{e}_{ik}}{\partial \{\mathbf{P}_{wk}, \mathbf{Q}_{wk}\}} & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \frac{\partial \mathbf{e}_{ij}}{\partial \mathbf{X}_{wj}}^\top \boldsymbol{\Sigma}_{\mathbf{e}_{ij}}^{-1} \frac{\partial \mathbf{e}_{ij}}{\partial \boldsymbol{\xi}_{iw}} & \vdots & \vdots & \frac{\partial \mathbf{e}_{ij}}{\partial \mathbf{X}_{wj}}^\top \boldsymbol{\Sigma}_{\mathbf{e}_{ij}}^{-1} \frac{\partial \mathbf{e}_{ij}}{\partial \mathbf{X}_{wj}} & \vdots & \vdots & \mathbf{0} & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \frac{\partial \mathbf{e}_{ik}}{\partial \{\mathbf{P}_{wk}, \mathbf{Q}_{wk}\}}^\top \boldsymbol{\Sigma}_{\mathbf{e}_{ik}}^{-1} \frac{\partial \mathbf{e}_{ik}}{\partial \boldsymbol{\xi}_{iw}} & \vdots & \vdots & \mathbf{0} & \vdots & \vdots & \frac{\partial \mathbf{e}_{ik}}{\partial \{\mathbf{P}_{wk}, \mathbf{Q}_{wk}\}}^\top \boldsymbol{\Sigma}_{\mathbf{e}_{ik}}^{-1} \frac{\partial \mathbf{e}_{ik}}{\partial \{\mathbf{P}_{wk}, \mathbf{Q}_{wk}\}} & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (3.35)$$

It is important to remark that each observation error  $\mathbf{e}_{ij}$  or  $\mathbf{e}_{ik}$ , only depends on a single KF  $\boldsymbol{\xi}_{iw}$ , and a single landmark  $\mathbf{X}_{wj}$  or  $\{\mathbf{P}_{wk}, \mathbf{Q}_{wk}\}$ . Hence, the Hessian matrix can be formed by adding the influence of each observation to its corresponding block, as showed in Equation (3.35), where the contribution of two single features to the Hessian is presented (the full matrix is formed by  $\mathbf{H} = \sum_{i \in \mathcal{K}_l} \sum_{j \in \mathcal{P}_l} \sum_{k \in \mathcal{L}_l} \mathbf{H}_{i,jk}$ ).

Notice that, for the rest of observations that belong to the KFs that are not part of the local map, their Jacobian matrices  $\frac{\partial \mathbf{e}_{ij}}{\partial \boldsymbol{\xi}_{iw}}$  and  $\frac{\partial \mathbf{e}_{ik}}{\partial \boldsymbol{\xi}_{iw}}$  are equal to zero, since here we only optimize the local map while the rest of the KFs remain fixed.

It should also be underlined that in (3.30) the influence of the errors in both points and lines is weighted with  $\boldsymbol{\Sigma}_{\mathbf{e}_{ij}}^{-1}$  and  $\boldsymbol{\Sigma}_{\mathbf{e}_{ik}}^{-1}$ , respectively, which stand for the inverses of the covariance matrixes associated to the uncertainty of each projection error. In practice, though, it is more effective to set such covariances to the identity matrix and follow a similar approach to the one described in Section 3.F.4 as it introduces robust weights and also deals with the presence of outlier observations.

Finally, we remove from the map those landmarks with less than 3 observations, as they are less meaningful.

### 3.F.6 Loop Closure

In this work, we adopt a bag of words (BoW) approach based on the binary descriptors extracted for both the keypoints and the line segments in order to robustly cope with data association and loop closure detection.

In short, the BoW technique consists in summarizing all the information extracted from an image (in our proposal, the descriptors of keypoints and line segments) into a *word* vector, employing for that a vocabulary that has been built off-line from different image datasets. Then, as the camera moves, the words computed from the grabbed images are stored in a database that is later employed to search for the most similar image to the current keyframe.

In the following, we first address the process of detecting loop closures from the created BoWs, and then describe the correction of the pose estimations of the keyframes involved in the loop.

## Loop Closure Detection

The detection of loop closures involves both to find an image similar to the one being currently processed and to estimate the relative pose change between them, as described next.

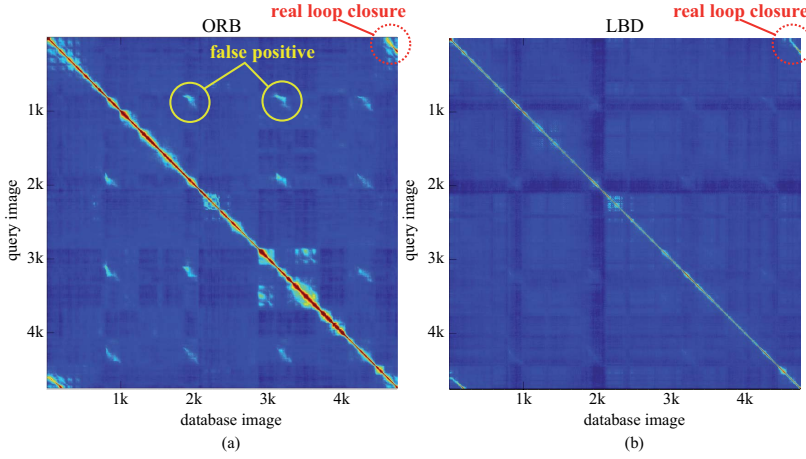
### 3.F.6.1 Visual Place Recognition

Specifically, we have employed the method presented in [57], which was initially developed for BRIEF binary descriptors, and subsequently adapted to ORB keypoints. Since, in our work, segments are also augmented with binary descriptors, we propose to build two specific visual vocabularies and databases for them. This way, at each time step, the most similar images in the databases of keypoints and line segments are retrieved in parallel in order to look for loop closures. This dual-search is motivated by the fact that some scenes may be described more distinctively by lines than by keypoints or vice versa. Thus, employing both methods and merging their results allow us to refine the output of database queries, incurring in a small computational footprint.

To illustrate this, we first define a *similarity matrix* as the matrix that contains in each row the similarity values, in the range  $[0,1]$ , of a certain image with all the images stored in the database. Then, we compute such matrices from a sequence recorded in a corridor that goes around a square area.

Concretely, the matrix in Figure 3.20(a) has been computed employing only ORB keypoints to build both the vocabulary and the database, while the one in Figure 3.20(b) relies only on lines. The color palette goes from blue (score = 0) to red (score = 1). As can be noted, some yellowish areas appear in the first matrix in places where the images look similar according to the keypoints (specifically, after turning at the corners of the corridor). This indicates potential loop closures although, in fact, they are just false positives. The second (line-only) matrix, though, does not present this behavior so that it may be employed to discard them. On the other hand, the first matrix presents more distinctiveness, since the difference in score is generally larger for non-similar images than in the line-only matrix. Therefore, the image similarities yielded by querying both feature databases may be combined to improve robustness when detecting potential loop closures.

In this work, we propose to weight the results from both features ( $s_k$  for keypoints and  $s_l$  for lines) according to two criteria, namely *strength* and *dispersion*. The former weights the similarity score proportionally to the number of features of a certain type (keypoint or line) in the set of features detected in the image, while the latter takes into account the dispersion of the features



**Figure 3.20:** Similarity matrices for a certain dataset where the (a) ORB keypoint-only bag-of-words approach yields false positives that are not present in the (b) LBD line-only approach.

in the image (the more disperse the higher the weight will be). This yields a more robust total similarity score for the image ( $s_t$ ):

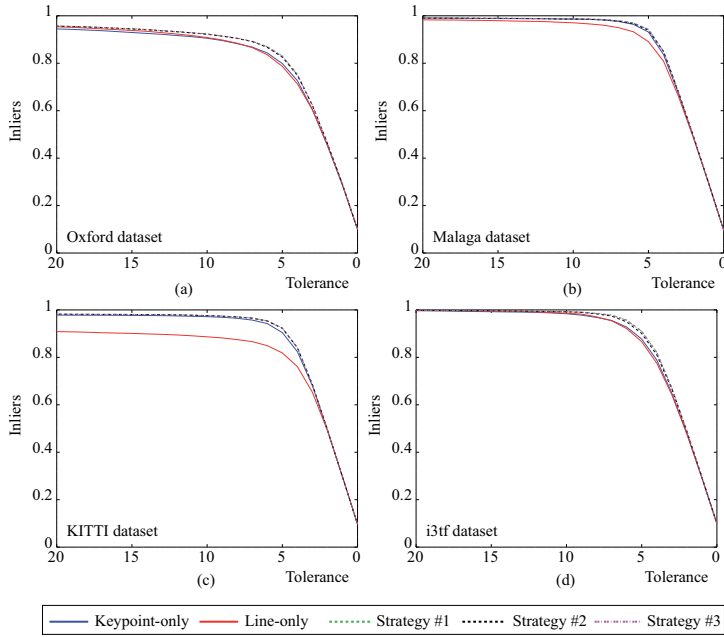
$$s_t = 0.5 (n_k / (n_k + n_l) + d_k / (d_k + d_l)) s_k + 0.5 (n_l / (n_k + n_l) + d_l / (d_k + d_l)) s_l. \quad (3.36)$$

In this equation,  $n_k$  and  $n_l$  are the number of keypoints and lines extracted in the image, respectively, while the dispersion values for the keypoints and the lines ( $d_k$  and  $d_l$ , respectively) are computed as the square root of the sum of the variances in the  $x$  and  $y$  coordinates of the found features. For the case of the lines, such  $x$  and  $y$  coordinates are taken from their midpoint.

Note that this formulation gives the same importance to both kinds of features (hence the 0.5 factor), although this could be tuned according to the environment (e.g. if the images are expected to be low-textured, it might be more convenient to down-weight the keypoint result with respect to the lines one). Nevertheless, the results will not change significantly with the weighting factor, and a finer optimization would not be worthy as long as both kind of features have influence in the total score.

We have empirically evaluated this strategy in comparison to four other alternatives following the classification framework employed in [97] for four different datasets: Oxford dataset [133], sequence 4 in Malaga dataset [20], sequence 7 in KITTI dataset [58] and i3tf dataset [151]. Concretely, the compared alternatives consisted in taking into account:

- only the score yielded from querying the *keypoint* bag-of-words ( $s_k$ ),



**Figure 3.21:** Precision-recall curves for four different datasets: (a) Oxford dataset, (b) sequence 4 in Malaga dataset, (c) sequence 7 in KITTI dataset and (d) i3tf dataset, for the 10 most similar images in the dataset.

- only the score yielded from querying the *line* bag-of-words ( $s_l$ ),
- both  $s_k$  and  $s_l$  but following only the *strength* criterion (strategy #1),
- both  $s_k$  and  $s_l$  but following only the *dispersion* criterion (strategy #2).

We have tested them on synchronized sequences without any loop closures, so that the elements in the diagonal matrix are 1, as they correspond to the same image in both the database and the query. Subsequently, we have selected, for each query image, the  $k$  most similar images from the database (i.e. those with largest total score), and have considered a match as an inlier (true positive match) if it was close enough to the diagonal with a tolerance of  $d$  frames. Finally, we have varied the tolerance and measured the ratio of inliers for all the strategies to generate the precision-recall curves in the above mentioned datasets.

As shown in Figure 3.21, all three combined strategies outperform the point and line-only approaches, while strategy #3 (that corresponding to Equation 3.36) performs slightly better than the other two in all the evaluated datasets.

### 3.F.6.2 Estimating the Relative Motion

Once we have a loop closure candidate, we still need to discard false positives that could have not been detected with the above mentioned approach. This is achieved by recovering the relative pose between the two KFs involved in the loop closure (namely *current* and *old* KFs from now on). For that, we first look for matches between the features from both KFs in a similar way to the one described in Section 3.F.3, while also searching for new correspondences between the current KF and the local map associated to the *old* one. Then, we estimate a valid transformation  $\hat{\xi}_{ij} \in \mathfrak{se}(3)$  that relates both KFs following the approach described in Section 3.F.4. Finally, since an erroneous detection of a loop closure (false positive) would produce a very negative impact on the SLAM system, we check the consistency of the loop closure candidate with the following tests:

- i) The maximum eigenvalue of the covariance matrix  $\Sigma_{\hat{\xi}_{ij}}$  is inferior to 0.01.
- ii) The obtained translation and rotation cannot rise over 0.50 meters and 3.00 degrees, respectively.
- iii) The inliers ratio in the estimation is higher than 50%.

Regarding the first criterion, a large value of the eigenvalues of the uncertainty matrix (see (3.27)) is often an indicator of an ill-conditioned Hessian matrix, most probably due to the presence of a large number of outliers in the feature matching set. Ensuring that the maximum eigenvalue of the covariance matrix is below a certain threshold allows us to detect potentially incorrect loop closures candidates and discard them.

In the case of the second criterion, we also set a maximum translation and rotation limit for the estimated pose, as BoW-based approaches typically provides positive matches that are very similar in appearance and pose, so that a large change in pose between the involved frames usually indicates a wrong loop closure detection. Finally, the third criterion sets a minimum ratio of detected inliers after the optimization process, since motion estimation is strongly affected by the presence of outliers and incorrect associations from visual place recognition.

### Loop Correction

After estimating all consecutive loop closures in our trajectory, we then fuse both sides of the loop closure correcting the error distributed along the loop. This is typically solved by formulating the problem as a pose-graph optimization (PGO), where the nodes are the KFs inside the loop, and the edges are given by both the essential graph and the spanning tree. For that, let us define the following error function as the  $\mathfrak{se}(3)$  difference between the transformation

that relates the KFs  $\hat{\xi}_{ij}$  to the current observation of the same transformation:

$$\mathbf{r}_{ij}(\xi_{iw}, \xi_{jw}) = \log(\exp(\hat{\xi}_{ij}) \cdot \exp(\xi_{jw}) \cdot \exp(\xi_{iw})^{-1}) \quad (3.37)$$

where the operators  $\log : SE(3) \mapsto \mathfrak{se}(3)$  and  $\exp : \mathfrak{se}(3) \mapsto SE(3)$  refer to the well-known logarithm and exponential maps. Notice that in the case of a regular edge, the value of  $\hat{\xi}_{ij}$  coincides with the estimation of  $\xi_{ij}$  in the first step of the optimization, and hence the error in these edges is initially zero.

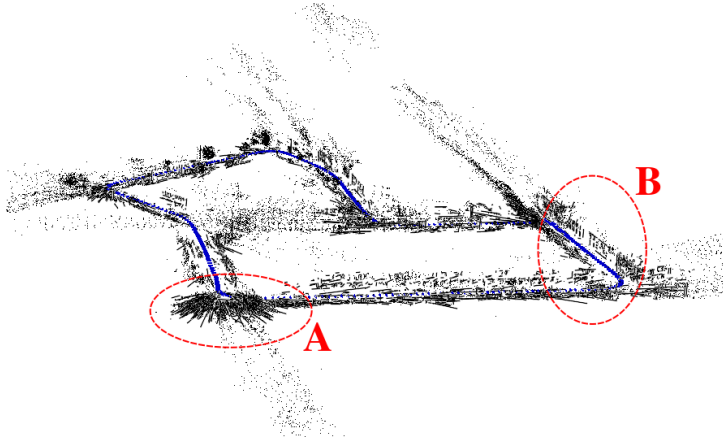
This PGO problem is solved using the `g2o` library [91] yielding the optimal pose of the KFs included in the optimization, i.e. the essential graph and the spanning tree, when considering the loop closure edges. Finally, we update the pose of the KFs along with the pose of the landmarks observed by them, and we also merge the local maps of both sides of the loop by first fusing the landmarks matched while estimating their relative motion (please, refer to Section 3.F.6), and then looking for new correspondences between the unmatched landmarks.

### 3.F.7 Experimental Validation

In this section we evaluate the performance of PL-SLAM in several video sequences from different datasets, in order to demonstrate the robustness of our proposal, which does not fail in any of the considered sequences, even in the low-textured ones. Besides, we also provide an estimation of both the camera trajectory and the map while computing a measurement of the committed error with respect to the ground truth. Concretely, we have tested our proposal against the EuRoC MAV dataset [23], a low-textured dataset specifically recorded for assessing the effect of adding lines to the visual SLAM system, and the well-known KITTI video sequences [58].

It is important to remark that the error metric employed in this paper for the EuRoC MAV and the low-textured datasets is the one proposed in [140] as Relative Pose Error (RPE), which computes the relative error in translation between the estimated camera poses and the ground truth. Using RPE as a metric allows us to obtain comparable error values for all the experiments, regardless the presence and number of loop closures in the camera trajectory, as relative measurements are not significantly affected by them, unlike absolute pose error measurements. For the KITTI dataset, though, we employ the standardized metrics provided by the KITTI benchmark to obtain the error measurements presented in Table 3.10.

In the following, we present examples of the trajectories and maps estimated by PL-SLAM, together with the average errors committed by (i) our proposal, (ii) a *point-only* version of our system (P-SLAM), (iii) a *line-only* version of our system (L-SLAM), and (iv) ORB-SLAM2, which is considered one of the state-of-the-art methods for stereo visual SLAM. For the latter, we have employed the open source implementation of its last version [116].



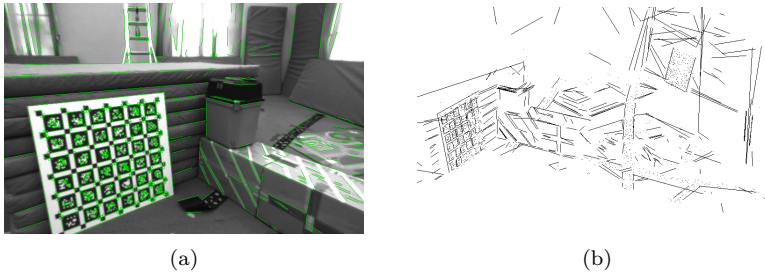
**Figure 3.22:** Map (in black) comprising points and line segments, and the trajectory (in blue) obtained with PL-SLAM from an outdoor environment in the sequence *KITTI-07*. The map presents noisy measurements in some parts (e.g. zone A), and structural lines from the environment, such as parts of the buildings (e.g. zone B).

We have also tried to compare our method against the one proposed in [151], but unfortunately, as their approach to perform line segment tracking is based on an optical flow algorithm, their proposal fails when applied to datasets with large motions between frames. Therefore, we could not include their results in this paper.

All the experiments have been run on an Intel Core i5-6600 CPU @ 3.30GHz and 16GB RAM without GPU parallelization.

### EuRoC MAV dataset

The EuRoC MAV dataset [23] consists of 11 stereo sequences recorded with a MAV flying across three different environments: two indoor rooms and one industrial scenario, containing sequences that present different challenges depending on the speed of the drone, illumination, texture, etc. As an example, we show the central part of the map built from the *V1-02-easy* sequence in Figure 3.23(b), where two different parts are clearly visible. The first one shows the features extracted from the non-structured part of the environment (refer to the right side of the map), presenting a relatively large amount of small and noisy line segments, which make difficult the interpretation of that part of the scene. In contrast, at the bottom left part of the figure, we can observe the structured part of the environment, which is clearly represented in the map through a set of line segments that depicts a checkerboard and a bunch of boxes. This example reflects that the maps built from line segments are



**Figure 3.23:** Mapping results in the *V1-01-easy* sequence from the EuRoC MAV dataset. (a) Features tracked between two consecutive keyframes. (b) Resulting 3D map for the sequence. The checkerboard and the boxes in the scene are clearly reflected in the left part of the map, while more noisy features can be found in the rest, as a consequence of factors like non-textured surfaces, high illumination, etc.

geometrically richer than those created from only points, so that they can be employed to extract high-level meaningful information from them.

Finally, Table 3.8 shows the mean relative translational RMSE of the motion estimation in the different sequences included in the dataset. It can be observed that, for indoor and structured scenarios, the inclusion of line segment features in the system is very beneficial to improve the robustness of the system and the estimation of the camera trajectory. In this case, both the *point-only* and the *line-only* approaches yield worse results than PL-SLAM, while ORB-SLAM2 fails in some sequences, as keypoint tracking is prone to be lost. PL-SLAM, on the contrary, successfully estimates the camera trajectory in all the sequences.

### Low-textured Scenarios

We have also assessed the performance of the compared methods in challenging low-textured scenarios. For that, we have recorded a set of four stereo sequences (namely *lt-easy*, *lt-medium*, *lt-difficult* and *lt-rot-difficult*) in a room equipped with an OptiTrack system<sup>1</sup>, which provides the ground-truth of the camera trajectory. The resulting covisibility graph yielded by our PL-SLAM system for the sequence *lt-medium* is shown in Figure 3.19, where a loop closure between the initial and the final part of the trajectory can be observed. The experiments in these sequences (refer to Table 3.9) reveal that, while point-based approaches either fail to recover the trajectory or yield worse results than in previous scenarios, the two methods based on line segments are capable of robustly estimating the camera path in all sequences, even achieving a good performance in terms of accuracy.

<sup>1</sup><http://optitrack.com/>

**Table 3.8:** Relative translational RMSE errors in the EuRoC MAV dataset [23]. A dash indicates that the experiment failed.

Sequence	P-SLAM	L-SLAM	PL-SLAM	ORB-SLAM2
MH-01-easy	0.0811	0.0588	0.0416	<b>0.0251</b>
MH-02-easy	0.1041	0.0566	<b>0.0522</b>	0.0638
MH-03-med	0.0588	<b>0.0371</b>	0.0399	0.0712
MH-04-dif	-	0.1090	0.0641	<b>0.0533</b>
MH-05-dif	0.1208	0.0811	0.0697	<b>0.0414</b>
V1-01-easy	0.0583	0.0464	0.0423	<b>0.0405</b>
V1-02-med	0.0608	-	<b>0.0459</b>	0.0617
V1-03-dif	0.1008	-	<b>0.0689</b>	-
V2-01-easy	0.0784	0.0974	<b>0.0609</b>	-
V2-02-med	0.0767	-	<b>0.0565</b>	0.0666
V2-03-dif	0.1511	-	<b>0.1261</b>	-

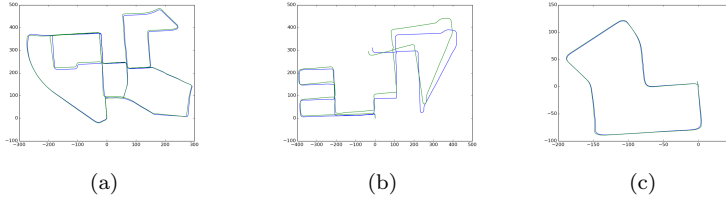
**Table 3.9:** Relative translational RMSE errors in low-textured sequences recorded with GT data from an OptiTrack system. A dash indicates that the experiment failed.

Sequence	P-SLAM	L-SLAM	PL-SLAM	ORB-SLAM2
lt-easy	-	0.1412	<b>0.1243</b>	0.1391
lt-medium	-	0.1998	<b>0.1641</b>	-
lt-difficult	-	0.1801	<b>0.1798</b>	-
lt-rot-difficult	0.2411	0.2247	<b>0.2034</b>	0.2910

### KITTI dataset

Finally, we have tested PL-SLAM on the well-known KITTI dataset [58], using the 11 sequences that provide ground truth and yielding the results presented in Table 3.10. Note that this is an urban dataset with highly textured image sequences and, as expected, the exploitation of line segments barely increases the accuracy, since point features are more than sufficient for a proper operation of a keypoint-based SLAM system. Still, the reasons why we have tested our proposal against the KITTI sequences are twofold: (i) the lack of publicly accessible datasets containing both low-textured scenes and ground truth, and (ii) the KITTI dataset has become a standard when assessing visual SLAM.

For this dataset, we have employed the standard error measurements proposed by the KITTI benchmark site, where the translational errors are expressed in % of the trajectory length while the rotational part is expressed in deg/100m of the trajectory. It is important to highlight that these results are obtained from applying the KITTI benchmarking scripts to the trajec-



**Figure 3.24:** Some trajectories estimated with PL-SLAM (in green) from the KITTI dataset (ground-truth in blue). (a) Trajectory estimated in the sequence *KITTI-00*, where a large amount of loop-closures can be found. (b) The sequence *KITTI-08* does not present any loop closure, and hence the drift along the trajectory is not corrected. (c) Finally, the sequence *KITTI-07* presents a loop closure between the initial and final parts of the trajectory.

ries estimated by the evaluated methods. This also includes ORB-SLAM2, for which we have generated the estimated trajectory for each sequence with both the code and the parameters they publicly provide in their repository. A different (and probably more tuned) set of parameters might cause the difference between their results obtained in this evaluation and those presented in the ORB-SLAM2 original paper, which could not be reproduced.

In any case, and regarding the robustness of the system, both PL-SLAM and ORB-SLAM2 successfully complete the trajectory estimation for all the sequences, as expected for such a textured dataset. Unsurprisingly, the results confirm worse performance of the *line-only* approach in these outdoor scenarios, even failing at properly estimating the trajectory of the stereo camera in some of the sequences (those recorded in rural environments, which lacks structural lines).

As an illustrative example, Figure 3.22 depicts the trajectory and the map estimated by PL-SLAM in the sequence *KITTI-07*. As can be seen in the zone marked as A in the figure, the presence of line segments can introduce some 'noise' in the maps, as not all the detected lines have a physical meaning, i.e. some lines do not belong to structural parts of the environment. Nevertheless, in other parts of the sequence, relevant information of the scene structure has been correctly captured in the map. This can be observed in the zone marked as B in the figure, where the buildings can be clearly observed, leading to a descriptive representation of the scene. On the contrary, the presence of noisy points in the map is less noticeable to the human eye, as they do not provide as much spatial information as line segments.

Finally, Figure 3.24 depicts the estimated trajectory obtained with PL-SLAM in three sequences from the KITTI dataset that present different number of loop closures. It can be noted the importance of correcting the drift in long sequences to obtain accurate absolute solutions (refer to Figure 3.24(a,c)),

**Table 3.10:** Mean relative RMSE for the KITTI dataset [58]. The translation errors are expressed in %, while the rotation errors are also expressed relatively to the translation in *deg/100m*. A dash indicates that the experiment failed.

Seq.	P-SLAM		L-SLAM		PL-SLAM		ORB-SLAM2	
	$t_{rel}$	$R_{rel}$	$t_{rel}$	$R_{rel}$	$t_{rel}$	$R_{rel}$	$t_{rel}$	$R_{rel}$
00	2.47	0.95	3.32	6.67	2.36	0.89	1.82	0.58
01	8.25	5.85	-	-	5.80	2.32	1.24	0.32
02	2.54	1.04	6.62	9.42	2.35	0.91	1.81	0.46
03	3.88	1.71	6.83	7.14	3.74	1.54	2.79	0.44
04	2.14	0.42	-	-	2.21	0.30	1.07	0.18
05	2.03	0.96	3.06	2.25	1.74	0.88	1.03	0.37
06	4.37	3.01	5.99	8.29	3.51	2.72	1.25	0.56
07	2.65	1.35	3.58	6.52	1.83	1.03	0.94	0.46
08	2.73	1.34	6.15	9.23	2.18	1.15	1.78	0.57
09	1.90	1.02	6.92	4.66	1.68	0.92	1.11	0.42
10	1.92	1.32	6.86	5.55	1.21	0.99	1.09	0.46

in contrast to the results obtained in sequences without loop closures, as the one presented in Figure 3.24(b).

### Performance

Regarding the time performance, we present Table 3.11 that shows the average processing time of each part of the PL-SLAM algorithm, for each of the tested datasets. Thanks to the efficient implementation of [62], our VO thread achieves real-time performance in average for all combinations of features (i.e. points, lines, and points and lines) and for all the datasets, since the acquisition time for the KITTI sequences is 10 fps and for both the EuRoC MAV and the *low-textured* datasets is 20 fps, while our proposal performs in 15 fps, 20 fps and 25 fps, respectively.

On the other hand, the local bundle adjustment (LBA) can be processed at around 20 Hz, which is fast enough for our purposes, as it runs in a parallel thread while the VO thread is continuously processing new frames. Finally, the *loop closure* management, although being the most time consuming step of the algorithm, presents acceptable values. Please, notice that these are average values, and, for each particular sequence, loop closure varies substantially: those presenting small loops can perform loop closure in few milliseconds while sequences with loops involving a significant amount of keyframes and landmarks spend higher processing time than either the local mapping or the visual odometry procedures. In any case, since loop closure is computed in a parallel thread (and not at every frame), the rest of the system can still run in real time.

**Table 3.11:** Average runtime of each part of the algorithm.

	KITTI 1241 × 376 10 fps	EuRoC MAV 752 × 480 20 fps	Low-Textured 752 × 480 20 fps
Visual Odometry			
P-SLAM	12.2 ms	8.7 ms	8.1 ms
L-SLAM	54.6 ms	47.6 ms	46.1 ms
PL-SLAM	66.0 ms	49.7 ms	40.0 ms
ORB-SLAM2	98.1 ms	69.0 ms	61.4 ms
Local Mapping			
P-SLAM	38.9 ms	37.3 ms	35.8 ms
L-SLAM	37.4 ms	36.0 ms	34.5 ms
PL-SLAM	43.8 ms	40.6 ms	42.1 ms
ORB-SLAM2	230.0 ms	162.0 ms	102.0 ms
Loop Closing			
P-SLAM	11.3 ms	3.5 ms	3.7 ms
L-SLAM	9.5 ms	3.9 ms	3.4 ms
PL-SLAM	28.0 ms	4.7 ms	4.5 ms
ORB-SLAM2	9.1 ms	3.6 ms	4.4 ms
Keyframe rate			
P-SLAM	4.78 f/kf	6.36 f/kf	3.28 f/kf
L-SLAM	3.68 f/kf	4.45 f/kf	3.25 f/kf
PL-SLAM	5.06 f/kf	5.36 f/kf	4.78 f/kf
ORB-SLAM2	2.77 f/kf	5.26 f/kf	3.10 f/kf

Finally, in order to illustrate the real-time capability of our system, we have also provided in Table 3.11 the rate of captured frames before inserting a new keyframe for each dataset, thus giving a picture of how often the system demands a local mapping and loop closure management.

## Discussion

The experimental validation presented in this paper proves that PL-SLAM operates more robustly than point-only approaches, specially in low-textured scenarios where keypoints are difficult to extract and track. Moreover, our implementation achieves real-time performance for all the considered datasets, which include an heterogeneous set of both indoor and outdoor scenarios.

Regarding the accuracy, we have measured the relative RMSE between keyframes for both the EuRoC MAV and the low-textured datasets, as relative measurements are not influenced by the presence of loop closures, hence

leading to results that can be more comparable between different sequences. On the other hand, we have employed the metrics proposed in the KITTI benchmark (i.e. absolute translational errors expressed in % of the trajectory length and absolute rotational errors expressed in deg/100m of the trajectory) in our experiments with the KITTI sequences, as they have become standardized for such dataset. PL-SLAM's results reveal better performance, in terms of accuracy, than its *point-only* and *line-only* approaches, and somewhat inferior performance of the ORB-SLAM2 method while, at the same time, providing more robustness in challenging, low-textured scenes in where point-only approaches are prone to fail. This inferior performance is mainly explained by the fact that our approach does not perform local bundle adjustment in every frame, unlike ORB-SLAM2, hence slightly increasing the final drift of the trajectory, especially in those sequences without loop closures.

Finally, and since our system architecture is strongly based on that of ORB-SLAM, we would like to highlight the essential differences between these two approaches: i) the inclusion of line segments as image features, which allows us to achieve robust camera localization in scenarios where keypoint-only methods usually perform poorly or even fail, ii) the inclusion of binary line descriptors in the loop closure procedure, in order to make it more robust, and iii) the implementation of the visual odometry thread as a frame-to-frame incremental motion estimation to meet the computational constraints that line segments introduce, unlike ORB-SLAM2, which continuously performs Local Bundle Adjustment between recent frames.

### 3.F.8 Conclusions

In this paper we have proposed a novel stereo visual SLAM system that extends our previous VO approach in [62], and that is based on the combination of both keypoints and line segment features. Our proposal, coined PL-SLAM, contributes with a robust and versatile system capable of working in all types of environments, including low-textured ones, while producing geometrically meaningful maps. For that, we have developed the first open source SLAM system that runs in real time and that simultaneously employs keypoints and line segment features. Our implementation has been developed from scratch and its based on a bundle adjustment solution that seamlessly deals with the combination of different kinds of features. Moreover, we have extended the place recognition bag-of-words approach in [57] for the case of simultaneously employing points and line segments, in order to enhance the loop-closure process. Our approach has been tested on well-known datasets such as EuRoC MAV or KITTI, as well as in a sequence of stereo images recorded in a challenging low-textured scenario. In these experiments, PL-SLAM has been compared to ORB-SLAM2 [116], a *point-only* system and a *line-only* system, obtaining superior performance in terms of robustness in most of the dataset sequences, while still operating in real-time.

With respect to the accuracy, our proposal gets similar results to ORB-SLAM2 for the EuRoC MAV and low-textured datasets when using the metric defined in [140], which computes the relative RMSE between keyframes. On the other hand, the experiments with the KITTI dataset shows somewhat superior performance to the ORB-SLAM2 method following its standardized metrics for both absolute translational and rotational errors.

For future work, our implementation can benefit from faster keypoint front-ends, such as the ones in SVO [53,54] and PL-SVO [61], where authors reduced the computational time of the feature tracking with a semi-direct approach that estimates the position of the features as a consequence of the motion estimation, or from alternative tracking techniques [63] to improve robustness in difficult illumination conditions. Finally, our algorithm can be employed to obtain more accurate and refined maps by applying some SfM or Multi-Stereo techniques [72,122] in order to filter the structural lines, hence obtaining more meaningful information of the structured parts of the environment.



## Dealing with Dynamic Illumination and HDR Environments

### 4.A Introduction

Despite the impressive results reached by state-of-art SLAM and VO algorithms in controlled lab environments, their robustness in more realistic scenarios is still an open challenge. While there are different challenges for robust VO and SLAM, as aforementioned, this chapter mainly focuses in improving the robustness to dynamic or challenging illuminations and HDR environments.

The difficulties in such environments come not only from the limitations of the sensors, conventional cameras often take over/under-exposed images in such scenes, but also from the bold assumptions of VO or SLAM algorithms, such as brightness constancy. These assumptions are severely violated when navigating in these situations, due to both the automatic adjustment of camera parameters (*e.g.* auto-exposure or HDR compensation) that results in global or local changes of image features or to the rapidly varying appearance variations occurring when traversing a dynamically illuminated scenario. As a consequence, the number of features successfully tracked in such sequences dramatically drops and therefore the localization problem becomes extremely difficult to be solved.

To overcome these difficulties, two recent research lines have emerged respectively: Active VO and Photometric VO. While the former tries to achieve robustness by externally controlling the camera parameters (gain or exposure time) [129] [155], the latter explicitly models the brightness change using the photometric model of the camera [96] [49].

## 4.B Contributions

The aforementioned approaches have demonstrated to improve robustness to challenging illuminations, however, they require advanced and detailed knowledge of the sensor and a heuristic setting of several parameters, which cannot be easily generalized. With the purpose of avoiding to deal with complex image models for dynamic illuminations or invasive hardware methods to actively control the parameter settings, the contributions of this thesis address this problem from two different perspectives.

The first contribution to this matter, presented in [66], addresses this problem from a deep learning perspective. For that, input images are enhanced to more informative and invariant representations for VO and SLAM thanks to the generalization properties of deep neural networks to achieve robust performance in varied conditions. In this work it is also demonstrated how the insertion of long short term memory allowed for temporally consistent sequences, as the estimation depends on previous states. The claims are validated by comparing the performance of two state-of-art algorithms in monocular VO/SLAM (ORB-SLAM [115] and DSO [49]) with the original input and the enhanced sequences, showing the benefits of this approach in challenging environments.

A more traditional perspective, purely geometrical, was exploited in [63] for the robust tracking of line segments for challenging stereo sequences with difficult illumination conditions. In this contribution we claimed that line segments can be successfully tracked by only considering their geometric consistency along consecutive frames, for which the tracking problem was stated as a *sparse, convex*  $\ell_1$ -minimization of the geometrical constraints from any line segment in the first image over all the candidates in the second one, within a *one-to-many* scheme. The claimed features are validated by evaluating both the matching performance and motion estimation in challenging video sequences from benchmarked datasets.

---

## 4.C Learning-based Image Enhancement for Visual Odometry in Challeng- ing HDR Environments

---

Ruben Gomez-Ojeda, Zichao Zhang, Javier Gonzalez-Jimenez,  
and Davide Scaramuzza

*Published in Proc. International Conference on Robotics and Automation  
(ICRA), 2018.*

©IEEE (Revised layout)

# Learning-based Image Enhancement for Visual Odometry in Challenging HDR Environments

*Ruben Gomez-Ojeda<sup>1</sup>, Zichao Zhang<sup>2</sup>, Javier Gonzalez-Jimenez<sup>1</sup>,  
and Davide Scaramuzza<sup>2</sup>*

*Machine Perception and Intelligent Robotics (MAPIR) Group, University of  
Malaga, Spain.*

*Robotics and Perception Group, Dep. of Informatics, University of Zurich,  
and Dep. of Neuroinformatics, University of Zurich and ETH Zurich,  
Switzerland.*

## Abstract

One of the main open challenges in visual odometry (VO) is the robustness to difficult illumination conditions or high dynamic range (HDR) environments. The main difficulties in these situations come from both the limitations of the sensors and the inability to perform a successful tracking of interest points because of the bold assumptions in VO, such as brightness constancy. We address this problem from a deep learning perspective, for which we first fine-tune a deep neural network with the purpose of obtaining enhanced representations of the sequences for VO. Then, we demonstrate how the insertion of long short term memory allows us to obtain temporally consistent sequences, as the estimation depends on previous states. However, the use of very deep networks enlarges the computational burden of the VO framework; therefore, we also propose a convolutional neural network of reduced size capable of performing faster. Finally, we validate the enhanced representations by evaluating the sequences produced by the two architectures in several state-of-art VO algorithms, such as ORB-SLAM and DSO.

## Supplementary Materials

A video demonstrating the proposed method is available at  
[https://youtu.be/NKx\\_zi975Fs](https://youtu.be/NKx_zi975Fs)

### 4.C.1 Introduction

In recent years, Visual Odometry (VO) has reached a high maturity and there are many potential applications, such as unmanned aerial vehicles (UAVs) and

augmented/virtual reality (AR/VR). Despite the impressive results achieved in controlled lab environments, the robustness of VO in real-world scenarios is still an unsolved problem. While there are different challenges for robust VO (e.g., weak texture [47] [61]), in this work we are particularly interested in improving the robustness in HDR environments. The difficulties in HDR environments come not only from the limitations of the sensors (conventional cameras often take over/under-exposed images in such scenes), but also from the bold assumptions of VO algorithms, such as brightness constancy. To overcome these difficulties, two recent research lines have emerged respectively: Active VO and Photometric VO. The former tries to provide the robustness by controlling the camera parameters (gain or exposure time) [129] [155], while the latter explicitly models the brightness change using the photometric model of the camera [96] [49]. These approaches are demonstrated to improve robustness in HDR environments. However, they require a detailed knowledge of the specific sensor and a heuristic setting of several parameters, which cannot be easily generalized to different setups.

In contrast to previous methods, we address this problem from a *Deep Learning* perspective, taking advantage of the generalization properties to achieve robust performance in varied conditions. Specifically, in this work, we propose two different Deep Neural Networks (DNNs) that enhance monocular images to more informative representations for VO. Given a sequence of images, our networks are able to produce an enhanced sequence that is invariant to illumination conditions or robust to HDR environments and, at the same time, contains more gradient information for better tracking in VO. For that, we add the following contributions to the state of the art:

- We propose two different deep networks: a very deep model consisting of both CNNs and LSTM, and another one of small size designed for less demanding applications. Both networks transform a sequence of RGB images into more informative ones, while also being robust to changes in illumination, exposure time, gamma correction, etc.
- We propose a multi-step training strategy that employs the down-sampled images from synthetic datasets, which are augmented with a set of transformations to simulate different illumination conditions and camera parameters. As a consequence, our DNNs are capable of generalizing the trained behavior to full resolution real sequences in HDR scenes or under difficult illumination conditions.
- Finally, we show how the addition of Long Short Term Memory (LSTM) layers helps to produce more stable and less noisy results in HDR sequences by incorporating the temporal information from previous frames. However, these layers increase the computational burden, hence complicating their insertion into a real-time VO pipeline.

We validate the claimed features by comparing the performance of two state-of-art algorithms in monocular VO, namely ORB-SLAM [115] and DSO [49], with the original input and the enhanced sequences, showing the benefits of our proposals in challenging environments.

## 4.C.2 Related Work

To overcome the difficulties in HDR environments, works have been done to improve the image acquisition process as well as to design robust algorithms for VO.

### Camera Parameter Configuration

The main goal of this line of research is to obtain the best camera settings (i.e., exposure, or gain) for image acquisition. Traditional approaches are based on heuristic image statistics, typically the mean intensity (brightness) and the intensity histogram of the image. For example, a method for autonomously configuring the camera parameters was presented in [117], where the authors proposed to setup the exposure, gain, brightness, and white-balance by processing the histogram of the image intensity. Other approaches exploited more theoretically grounded metrics. [100], employed the Shannon entropy to optimize the camera parameters in order to obtain more informative images. They experimentally proved a relation between the image entropy and the camera parameters, then selected the setup that produced the maximum entropy.

Closely related to our work, some researchers tried to optimize the camera settings for visual odometry. [129] defined an information metric, based on the gradient magnitude of the image, to measure the amount of information in it, and then selected the exposure time that maximized the metric. Recently, [155] proposed a robust gradient metric and adjusted the camera setting according to the metric. They designed their exposure control scheme based on the photometric model of the camera and demonstrated improved performance with a state-of-art VO algorithm [53].

### Robust Vision Algorithms

To make VO algorithms robust to difficult light conditions, some researchers proposed to use invariant representations, while others tried to explicitly model the brightness change. For feature-based methods, binary descriptors are efficient and robust to brightness changes. [115] used ORB features [126] in a SLAM pipeline and achieved robust and efficient performance. Other binary descriptors [95] [26] are also often used in VO algorithms. For direct methods, [13] incorporated binary descriptors into the image alignment process for direct VO, and the resulting system performed robustly in low light.

To model the brightness change, the most common technique is to use an affine transformation and estimate the affine parameters in the pipeline. [79] proposed an adaptive algorithm for feature tracking, where they employed an affine transformation that modeled the illumination changes. More recently, a photometric model, such as the one proposed by [38], is used to account for the brightness change due to the exposure time variation. A method to deal with brightness changes caused by auto-exposure was published in [96], reporting a tracking and dense mapping system based on a normalized measurement of the radiance of the image (which is invariant to exposure changes). Their method not only reduced the drift of the camera trajectory estimation, but also produced less noisy maps. [49] proposed a direct approach to VO with a joint optimization of both the model parameters, the camera motion, and the scene structure. They used the photometric model of the camera as well as the affine brightness transfer function to account for the brightness change. In [155], the authors also adapted a direct VO algorithm [53] with both methods and presented an experimental comparison of using the affine compensation and the photometric model of the camera.

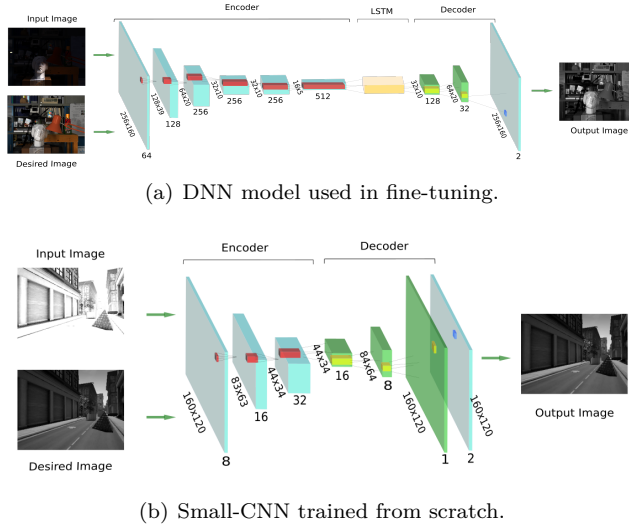
To the best of our knowledge, there is little work on using learning-based methods to tackle the difficulties in HDR environments. In the rest of the paper, we will describe how to design networks for this task, the training strategy and the experimental results.

### 4.C.3 Network Overview

In this work, we need to perform a pixel-wise transformation from monocular RGB images in a way that the outputs are still realistic images, on which we will further run VO algorithms. For pixel-wise transformation, the most used approach is DNNs structured in the so-called *encoder-decoder* form. These type of architectures have been successfully employed in many different tasks, such as optical flow estimation [42], image segmentation [81], depth estimation [103], or even to solve the image-to-image translation problem [75]. The proposed architectures (see Figure 4.1), implemented in the Caffe library [77], consist of an encoder, LSTM layers and a decoder, as described in the following.

#### Encoder

The encoder network consists of a set of purely convolutional layers that transform the input image, into a more reduced representation of feature vectors, suitable for a specific classification task. Due to the complexity of training from scratch [142], a standard approach is to initialize the model with the weights of a pre-trained model, known as *fine-tuning*. This has several advantages, as models trained with massive amount of natural images such as VGGNet [132], a seminal network for image classification, usually provide a good performance and stability during the training. Moreover, as initial layers closer to the input



**Figure 4.1:** Scheme of the architectures employed in this work. Both DNNs are formed by an *encoder* convolutional network, and a *decoder* that forms the enhanced output images. In the case of the fine-tuned network, we introduce a LSTM network to produce temporally consistent sequences. These figures have been adapted from [103,104].

image provide low-level information and final layers are more task-specific, it is also typical to employ the first layers of a well-trained CNN for different purposes, i.e. place recognition [64]. This was also the approach in [104], where authors employed the first 8 layers of VGGNet to initialize their network, keeping their weights fixed during training, while the remaining layers were trained from scratch with random initialization. Therefore, in this work, we first fine-tuned the very deep model in [104], depicted in Figure 4.1(a).

However, since our goal is to estimate the VO with the processed sequences, a very deep network, such as the fine-tuned model, is less suitable for usual robotic applications, where the computational power must be saved for the rest of modules. Moreover, depth estimation requires a high level of semantic abstraction as it needs some spatial reasoning about the position of the objects in the scene. In contrast, VO algorithms are usually based on tracking regions of interest in the images, which largely relies on the gradient, i.e., the first derivatives of the images, information that it is usually present in the shallow layers of CNNs. Therefore, we also propose a smaller and less deep CNN to obtain faster performance, whose encoder is formed by three layers (dimensions are in Figure 4.1(b)), each one of them formed by a convolution with a  $5 \times 5$  kernel, followed by a batch-normalization layer [74] and a pooling layer.

### Long Short Term Memory (LSTM)

While it is feasible to use a feedforward neural network to increase the information in images for VO, the input sequence may contain non-ignorable brightness variation. More importantly, the brightness constancy is not enforced in a feedforward network, hence the output sequence is expected to break the brightness constancy assumption for many VO algorithms. To overcome this, we can exploit the sequential information to produce more stable and temporally consistent images, i.e. reducing the impact of possible illumination change to ease the tracking of interest points. Therefore, we exploit the Recurrent Neural Networks (RNNs), more specifically, the LSTM networks first introduced in [71]. In these networks, unlike in standard CNNs where the output is only a non-linear function  $f$  of the current state  $\mathbf{y}_t = f(\mathbf{x}_t)$ , the output is also dependent on the previous output:

$$\mathbf{y}_t = f(\mathbf{x}_t, \mathbf{y}_{t-1}) \quad (4.1)$$

as the layers are capable of memorizing the previous states. We introduce two LSTM layers in the fine-tuned network between the encoder and the decoder part, in order to produce more stable results for a better odometry estimation.

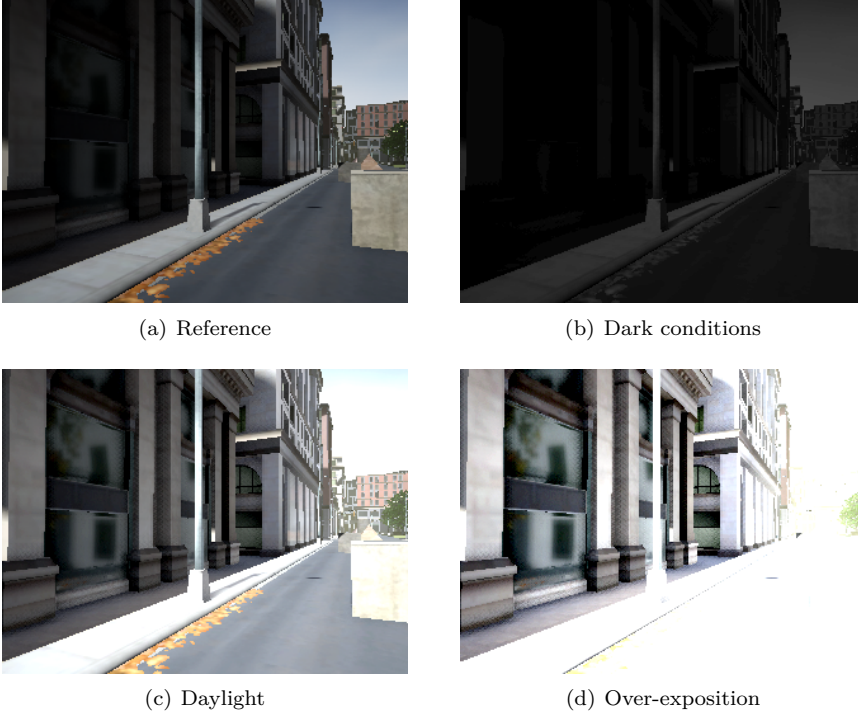
### Decoder

Finally, the decoder network is formed by three deconvolutional layers, each of them formed by an upsampling, a convolution and a batch-normalization layer, as depicted in Figure 4.1. The deconvolutional layers increase the size of the intermediate states and reduce the length of the descriptors.

Typically, decoder networks produce an output image of a proportional size of the input one containing the predicted values, which is in general blurry and noisy thus not very convenient to be used in a VO pipeline. To overcome this issue, we introduce an extra step which merges the raw output of the decoder with the input image producing a more realistic image. For that, we concatenate both the input image in grayscale and the decoder output into a 2-channel image then applying a final convolutional filter with a  $1 \times 1$  kernel and one channel.

#### 4.C.4 Training the DNN

Our goal is to produce an enhanced image stream to increase the robustness/accuracy of visual odometry algorithms under challenging situations. Unfortunately, there is no ground-truth available for generating the optimal sequences, nor direct measurement that indicates the goodness of an image for VO. To overcome this difficulties, we observe that the majority of the state-of-art VO algorithms, both *direct* and *feature-based* approaches, actually exploit the gradient information in the image. Therefore, we aim to train our network



**Figure 4.2:** Some training samples from the Urban dataset proposed in [104], for which we have simulated artificial illumination and exposure conditions by post-processing the dataset with different contrast and gamma levels.

to produce images containing more gradient information. In this section, we first introduce the dataset used for training then our training strategy.

### Datasets

To train the network, we need images taken at the same pose but with different illuminations, which are unfortunately rarely available in real-world VO datasets. Therefore we employed synthetic datasets that contain changes in the illumination of the scenes. In particular, we used the well-known New University of Tsukuba dataset [120] and the Urban Virtual dataset generated by [104], consisting of several sequences from an artificial urban scenario with non-trivial 6-DoF motion and different illumination conditions. In order to increase the amount of data, we simulated 12 different camera and illumination conditions (see Figure 4.2) by using several combinations of Gamma and Contrast values. Notice that this data augmentation must contain an equally distributed amount of conditions, otherwise the output of the network might

be biased to the predominant case. To select the best image  $\mathbf{y}^*$  (with the most gradient information), we use the following gradient information metric:

$$g(\mathbf{y}) = \sum_{u_i} \|\nabla \mathbf{y}(u_i)\|^2 \quad (4.2)$$

which is the sum of the gradient magnitude over all the pixels  $u_i$  in the image  $\mathbf{y}$ . For training the CNN, we used RGB images of  $256 \times 160$  pixels in the case of fine-tuning the model in [104] and grayscale images of  $160 \times 120$  pixels for the reduced network. We trained the LSTM network with full-resolution images ( $752 \times 480$ ) as, unlike convolutional layers, once trained they cannot be applied to inputs of different size.

### Training the CNN

We first train without LSTM, with the aim of obtaining a good CNN (*encoder-decoder*) capable of estimating the enhanced images from individual (not sequential) inputs. This part of training consists of two stages:

#### 4.C.4.1 Pre-training the Network

In order to obtain a good and stable initialization, we first train the CNN with pairs of images at the same pose, consisting of the reference image  $y^*$  and an image with different appearance. On our first attempts, we tried to optimize directly the bounded increments of the gradient information (4.2). The results are very noisy, due to the high complexity of the pixel-wise prediction problem. Instead, we opted to train the CNN by imposing the output to be similar to the reference image, in a pixel-per-pixel manner. For that, we employed the logarithmic RMSE, which is defined for a given reference  $y^*$  and an output  $y$  image as:

$$\mathcal{L}(\mathbf{y}, \mathbf{y}^*) = \sqrt{\frac{1}{N} \sum_i \|\log y_i - \log y_i^*\|^2}, \quad (4.3)$$

where  $i$  is the pixel index in the images. Although we tried different strategies for this purpose, such as the denoising autoencoder [145], we found this loss function much more suitable for VO applications, as it produced a smoother result than the Euclidean RMSE, specially for bigger errors, hence easing the convergence process. This first part of the training was performed with the Adam solver [85], with a learning rate  $l = 0.0001$  for 20 epochs of the training data, and a dataset formed by 80k pairs and requiring about 12 hours on a NVIDIA GeForce GTX Titan.

#### 4.C.4.2 Imposing Invariance

Once a good performance with the previous training was achieved, we trained the CNN to obtain invariance to different appearances. The motivation is that,

for images with different appearances (i.e. brightness) taken at the same pose, the CNN should be able produce the same enhanced image. For that, we selected triplets of images from the Urban dataset, by taking the reference image  $\mathbf{y}^*$ , and another two images  $\mathbf{y}_1$  and  $\mathbf{y}_2$  from the same place with two different illuminations. Then, we trained the network in a siamese configuration, for which we again imposed both outputs to be similar to the reference one. In addition, we introduced the following loss function:

$$\mathcal{L}_{SSIM}(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}^*) = SSIM(\mathbf{y}_1, \mathbf{y}_2) \quad (4.4)$$

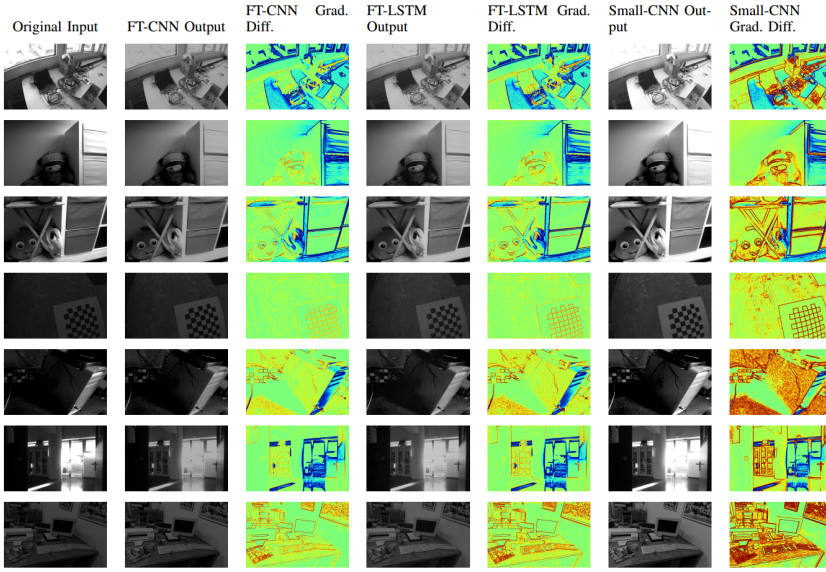
which is the structural similarity (SSIM) [148], usually employed to measure how similar two images are. This second part of the training was performed, during 10 epochs of the training data (40k triplets), requiring about 6 hours of training with the same parameters as in previous Section.

### Training the LSTM network

After we obtain a good CNN, the second part of the training is designed to increase the stability of the outputs, given that we are processing sequences of consecutive images. The goal is to provide not only more meaningful images, but also fulfill the brightness constancy assumption. For that purpose, we trained the whole DNN, including the LSTM network, with sequences of two consecutive images (i.e., taken at consecutive poses on a trajectory) under slightly different illumination conditions, while the reference ones presented the same brightness. The loss function consists of the LogRMSE loss function (4.3) to ensure that both outputs are similar to their respective reference ones, and the SSIM loss (4.4) without the structural term (as images do not belong to the exact same place) between the two consecutive outputs to ensure that they have a similar appearance. The LSTM training was performed during 10 epochs of the data (40k triplets), in about 12 hours with the same parameters as in previous Section.

## 4.C.5 Experimental Validation

In this section, we evaluate the performance of our approach by measuring two different metrics: the increments of gradient magnitude in the processed images and the improvements in accuracy and performance of ORB-SLAM [115] and DSO [49], two state-of-art VO algorithms for both *feature-based* and *direct* approaches, respectively. For that, we first run the VO experiments with the original image sequence, several standard image processing approaches, i.e. Normalization (N), Global Histogram Equalization (G-HE) [127], and Adaptive Histogram Equalization (A-HE) [158]. Then, we also evaluate the VO algorithms with the image sequences produced with the trained networks: the fine-tuned approaches FT-CNN and FT-LSTM, and the reduced model trained



**Figure 4.3:** Outputs from the trained models and difference between the gradient images in some challenging samples extracted from the evaluation sequences (the scale for the *jet* colormap remains fixed for each row).

from scratch Small-CNN. Notice that, even though the CNN networks proposed in this paper (not the FT-LSTM) have been trained only with synthetic images with reduced size ( $256 \times 180$  and  $160 \times 120$  pixels for the fine-tuned and our proposal respectively), the experiments have been performed with full-resolution ( $752 \times 480$  pixels) and real images.

### Gradient Inspection

As stated before, one way of measuring the quality of an image is its amount of gradient. Unfortunately, there is no standard metric for measuring the gradient information; actually, it is highly dependent on the application. In the case of visual odometry, it is even more important, as most approaches are based on edge information (which is directly related to the gradient magnitude image). Figure 4.3 presents the estimated images and the difference between the gradients of the output and the input images for several images from the trained models in different datasets. For the representation we have used the colormap *jet*, i.e. from blue to red, with  $\pm 30$  units of range (negative values indicate a decrease of the gradient amount). In general, we observe a general tendency in all models to reduce the gradient amount in the most exposed parts of the camera as they are less informative due to the sensor saturation, while increasing the gradient in the rest of the image.

### Evaluation with state-of-art VO algorithms

**Table 4.1:** ORB-SLAM [115] average RMSE errors (% first row) normalized by the length of the trajectory and percentage of the sequence without loosing the tracking (second row). A dash means that the VO experiment failed without initializing.

Dataset	ORB-SLAM [115]	N	G-HE	A-HE	FT-CNN	FT-LSTM	Small-CNN
<i>1-light</i>	3.91	4.07	-	-	3.52	<b>3.49</b>	4.62
	24.80	26.98	-	-	23.84	25.32	<b>80.52</b>
<i>2-lights</i>	2.19	2.17	-	2.27	<b>2.07</b>	2.09	2.72
	68.92	68.76	-	65.88	70.94	<b>72.98</b>	68.76
<i>3-lights</i>	3.78	3.81	-	3.63	<b>3.52</b>	3.81	3.65
	100.00	100.00	-	100.00	100.00	100.00	100.00
<i>switch</i>	3.60	4.85	-	4.56	5.64	<b>2.66</b>	2.97
	13.76	24.98	-	8.84	7.32	<b>31.02</b>	21.62
<i>hdr1</i>	5.67	5.67	3.71	-	5.22	5.21	<b>4.77</b>
	74.30	76.6	49.36	-	<b>81.54</b>	81.14	78.76
<i>hdr2</i>	3.49	4.08	4.42	3.52	<b>3.42</b>	3.88	3.51
	74.86	70.50	34.12	25.3	74.52	71.02	<b>75.22</b>
<i>overexposed</i>	2.64	2.57	2.59	<b>2.53</b>	2.72	2.65	2.83
	100.00	100.00	100.00	100.00	100.00	100.00	100.00
<i>bright-switch</i>	3.13	3.08	2.03	3.10	<b>1.97</b>	2.02	1.95
	34.60	34.94	100.00	35.42	100.00	100.00	100.00
<i>low-texture</i>	-	-	-	-	<b>5.28</b>	-	-
	-	-	-	-	<b>39.08</b>	-	-

**Table 4.2:** DSO [49] average RMSE errors normalized by the length of the trajectory for each method and trained network when evaluating. A dash means that the VO experiment failed.

Dataset	DSO [49]	N	G-HE	A-HE	FT-CNN	FT-LSTM	Small-CNN
<i>1-light</i>	2.39	-	2.37	2.42	<b>2.36</b>	<b>2.36</b>	2.40
<i>2-lights</i>	2.12	-	<b>2.05</b>	2.12	2.12	2.15	2.14
<i>3-lights</i>	<b>2.65</b>	-	2.66	2.66	2.66	2.69	2.69
<i>switch</i>	-	-	-	-	4.38	4.39	<b>2.90</b>
<i>hdr1</i>	2.46	4.80	2.34	2.52	2.42	<b>2.17</b>	2.44
<i>hdr2</i>	1.28	-	1.59	3.17	1.23	<b>1.22</b>	2.57
<i>overexposed</i>	1.61	1.60	1.64	1.62	<b>1.58</b>	<b>1.58</b>	1.60
<i>bright-switch</i>	4.51	-	1.49	1.47	1.93	<b>1.73</b>	4.43
<i>low-texture</i>	3.22	2.67	2.76	3.22	3.22	<b>3.14</b>	3.21

In order to evaluate the trained models in challenging conditions, we recorded 9 sequences with a hand-held camera in a room equipped with an OptiTrack system that allows us to also record the ground-truth trajectory of the camera and evaluate quantitatively the results. Each sequence was recorded for several illumination conditions: first with 1 – 3 lights available in the room, then without any light, and finally by switching the lights on and off during the sequence. It is worth noticing that, despite the numerous public benchmarks

**Table 4.3:** Average runtime and memory usage for each network

DNN	Res. (pixels)	Memory	GPU
FT-CNN	$256 \times 180$	371 MiB	23.80 ms
FT-CNN	$756 \times 480$	1175 MiB	149.72 ms
FT-LSTM	$756 \times 480$	3897 MiB	275.24 ms
Small-CNN	$160 \times 120$	135 MiB	<b>4.77 ms</b>
Small-CNN	$756 \times 480$	373 MiB	<b>48.4 ms</b>

available for VO, they are usually recorded in good and static illumination conditions, therefore our approach barely improves the trajectory estimation.

Table 4.1 shows the results of ORB-SLAM in all the sequences mentioned above. Firstly, we observe the benefits of our approach as our methods clearly outperform the original input and the standard image processing approaches in the difficult sequences (*1-light* and *switch*), while also maintaining a similar performance in the easy ones (*2-lights* and *3-lights*). As for the different networks, we clearly observe the better performance of FT-LSTM in the difficult sequences, although the reduced approach Small-CNN reports a good performance in the scene with the switching lights.

The results obtained with DSO are represented in Table 4.2. Since all the methods were successfully tracked, we omit the tracking percentage. In terms of accuracy, we again observe the good performance of the reduced approach, Small-CNN, with the direct approach. However, its accuracy is worse in the *bright-switch* sequence but it still performs similar to the original sequence.

### Computational Cost

Finally, we evaluate the computational performance of the two trained networks. For that, we compare the performance of the CNN and the LSTM, for both the training and the runtime image resolutions. All the experiments were run on a Intel(R) Core(TM) i7-4770K CPU @ 3.50GHz and 8GB RAM, and an NVIDIA GeForce GTX Titan (12GB). Table 4.3 shows the results of each model and all possible resolutions. We first observe that while obtaining comparable results to the fine-tuned model, the small CNN can perform faster (a single frame processing takes 3 times less than with FT-CNN and up to 5 times less than FT-LSTM for the resolution  $756 \times 480$ ), and therefore is the closest configuration to a direct application in a VO pipeline. It is also worth noticing the important impact of the LSTM layers in the performance, because they not only require a high computational burden but also double the size of the encoder network (a consecutive image pair is needed).

### 4.C.6 Conclusions

In this work, we tackled the problem of improving the robustness of VO systems under challenging conditions, such as difficult illuminations, HDR environments, or low-textured scenarios. For that, we solved the problem from a deep learning perspective, for which we proposed two different architectures, a very deep model that is capable of producing temporally consistent sequences due to the inclusion of LSTM layers, and a small and fast architecture more suitable for VO applications. We propose a multi-step training employing only reduced images from synthetic datasets, which are also augmented with a set basic transformations to simulate different illumination conditions and camera parameters, as there is no ground-truth available for our purposes. We then compare the performance of two state-of-art algorithms in monocular VO, ORB-SLAM [115] and DSO [49], when using the normal sequences and the ones produced by the DNNs, showing the benefits of our proposals in challenging environments.

---

## 4.D Geometric-based Line Segment Tracking for HDR Stereo Sequences

---

Ruben Gomez-Ojeda, Javier Gonzalez-Jimenez

*Published in Proc. International Conference on Intelligent Robots and Systems (IROS), 2018.*

©IEEE/RSJ (Revised layout)

# Geometric-based Line Segment Tracking for HDR Stereo Sequences

*Ruben Gomez-Ojeda, Javier Gonzalez-Jimenez*

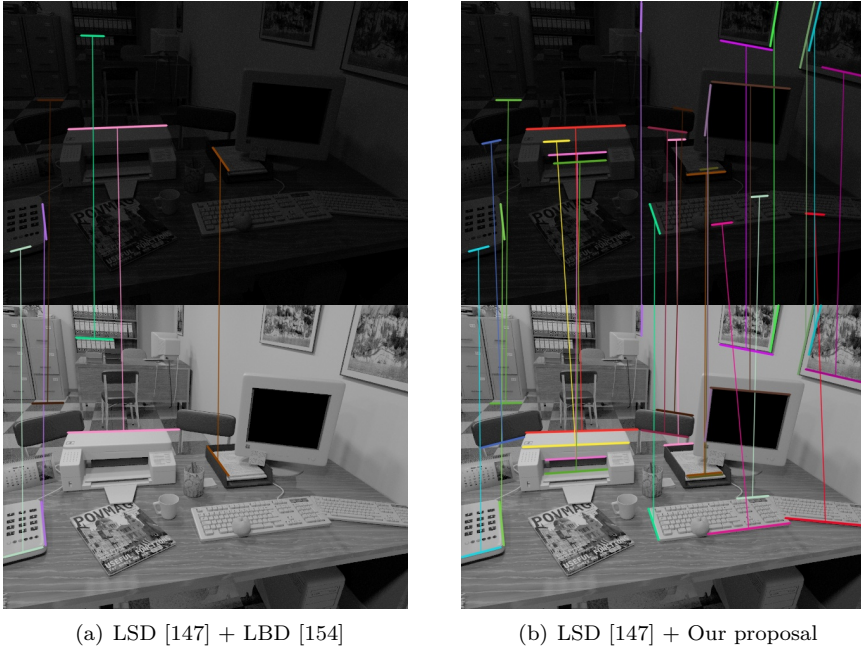
## Abstract

In this work, we propose a purely geometrical approach for the robust matching of line segments for challenging stereo streams with severe illumination changes or High Dynamic Range (HDR) environments. To that purpose, we exploit the univocal nature of the matching problem, i.e. every observation must be corresponded with a single feature or not corresponded at all. We state the problem as a sparse, convex,  $\ell_1$ -minimization of the matching vector regularized by the geometric constraints. This formulation allows for the robust tracking of line segments along sequences where traditional appearance-based matching techniques tend to fail due to dynamic changes in illumination conditions. Moreover, the proposed matching algorithm also results in a considerable speed-up of previous state of the art techniques making it suitable for real-time applications such as Visual Odometry (VO). This, of course, comes at expense of a slightly lower number of matches in comparison with appearance-based methods, and also limits its application to continuous video sequences, as it is rather constrained to small pose increments between consecutive frames. We validate the claimed advantages by first evaluating the matching performance in challenging video sequences, and then testing the method in a benchmarked point and line based VO algorithm.

### 4.D.1 Introduction

Although appearance-based tracking has reached a high maturity for *feature-based* motion estimation, its robustness in real-world scenarios is still an open challenge. In this work, we are particularly interested in improving the robustness of visual feature tracking in sequences including severe illumination changes or High Dynamic Range (HDR) environments (see Figure 4.4). Under these circumstances, traditional descriptors based on local appearance, such as ORB [126] and LBD [154] for points and line segments, respectively, tend to provide many outliers and a low number matches, and hence jeopardizing the performance of the visual tracker.

We claim that line segments can be successfully tracked along video sequences by only considering their geometric consistency along consecutive



**Figure 4.4:** Pair of consecutive frames extracted from the sequence *hdr/flicker1* from the dataset in [96] under challenging illumination changes. Our approach allows for the robust tracking of line segments in this type of environments, where traditional appearance-based matching techniques tend to fail.

frames, namely, the oriented direction in the image, the overlap between them, and the epipolar constraints. To achieve robust matches from this reduced segment description we need to introduce some mechanism to deal with the ambiguity associated to such purely geometrical line matching.

For that, we state the problem as a *sparse, convex*  $\ell_1$ -minimization of the geometrical constraints from any line segment in the first image over all the candidates in the second one, within a *one-to-many* scheme. This formulation allows for the successful tracking of line segments as it only accepts matches that are guaranteed to be globally unique. In addition, the proposed method results in a considerable speed-up of the tracking process in comparison with traditional appearance based methods. For this reason we believe this method can be a suitable choice for motion estimation algorithms intended to work in challenging environments, even as a recovery stage when traditional descriptor-based matching fails or does not provide enough correspondences.

To deal with outliers we impose some requiring constraints (e.g. small baseline between the two consecutive images) which slightly reduce the effectiveness

of line matching in scenes with repetitive structures and also the number of tracked features.

In summary, the contributions of this paper are the following:

- A novel technique for the tracking of line segments along continuous sequences based on a *sparse, convex*  $\ell_1$ -minimization of geometrical constraints, hence allowing for robust matching under severe appearance variations (see Figure 4.4).
- A efficient implementation of the proposed method yielding a less computationally demanding line-segment tracker which reduces one of the major drawbacks of working with these features.
- Its validation in our previous point and line features stereo visual odometry system [62], resulting in a more robust VO system under difficult illumination conditions, and also reducing the computational burden of the algorithm.

These contributions are validated with extensive experimentation in several datasets from a wide variety of environments, where we first compare the accuracy and precision of the proposed tracking technique, and then show its performance alongside a VO framework.

#### 4.D.2 Related Work

Feature-based motion reconstruction techniques, e.g. VO, visual SLAM, or SfM, are typically addressed by detecting and tracking several geometrical features (over one or several frames) and then minimizing the reprojection error to recover the camera pose. In this context, several successful approaches have been proposed, such as PTAM [88], a monocular SLAM algorithm that relies on FAST corners and SSD search over a predicted patch in a coarse-to-fine scheme for feature tracking. More recently, ORB-SLAM [116] contributed with a very efficient and accurate SLAM system based on a very robust local bundle adjustment stage thanks to its fast and continuous tracking of keypoints for which they relied on ORB features [126]. Unfortunately, even-though binary descriptors are relatively robust to brightness changes, these techniques suffer dramatically when traversing poorly textured scenarios or severe illumination changes occur (see Figure 4.4), as the number of tracked features drops.

Some works try to overcome the first situation by combining different types of geometric features, such as edges [47], edgelets [54], lines [18], or planes [102]. The emergence of specific line-segment detectors and descriptors, such as LSD [147] and LBD [154] allowed to perform feature tracking in a similar way as traditionally done with keypoints. Among them, in [90] authors proposed a stereo VO algorithm relying on image points and segments for which they implement a stereo matching algorithm to compute the

disparity of several points along the line segment, thus dealing with partial occlusions. In [62] we contribute with a stereo VO system (PLVO) that probabilistically combines ORB features and line segments extracted and matched with LSD and LBD by weighting each observation with their inverse covariance. In the SLAM context, the work in [151] proposes two different representations: Plücker line coordinates for the 3D projections, and an orthonormal representation for the motion estimation, however, they track features through an optical flow technique, thus the performance with fast motion sequences deteriorates. Unfortunately, the benefits of employing line segments come at the expense of higher difficulties in dealing with them (and they require a high computational burden in both detection and matching stages), and, more importantly, they still suffer from the same issues as keypoints when working with HDR environments.

A number of methods for dealing with varying illumination conditions have been reported. For example, [49] proposed a direct approach to VO, known as DSO, with a joint optimization of both the model parameters, the camera motion, and the scene structure. They used the photometric model of the camera as well as the affine brightness transfer function to account for the brightness change. In [155] authors contributed a robust gradient metric and adjusted the camera setting according to the metric. They designed their exposure control scheme based on the photometric model of the camera and demonstrated improved performance with a state-of-art VO algorithm [53]. Recently, [66] proposed a deep neural network that embeds images into more informative ones, which are robust to changes in illumination, and showed how the addition of LSTM layers produces more stable results by incorporating temporal information to the network. Although those approaches have proven to be effective to moderate changes in illumination or exposure, they would still suffer in more challenging scenarios such as the one in Figure 4.4.

### 4.D.3 Geometric-based Line Segment Tracking

#### Problem Statement

The first stage of our segment matching algorithm takes as input a pair of images from a stereo video sequence,  $I_1$  and  $I_2$ , which can be either from the stereo pair or two consecutive ones in the sequence. Let us define the sets of line segments  $\mathcal{L}_1 = \{\mathbf{s}_i, \mathbf{e}_i \mid i \in 1, \dots, m\}$  and  $\mathcal{L}_2 = \{\mathbf{s}_j, \mathbf{e}_j \mid j \in 1, \dots, n\}$  in  $I_1$  and  $I_2$ , where we represent the line segment  $k$  by their endpoints  $\mathbf{s}_k$  and  $\mathbf{e}_k$  in homogeneous coordinates. We also employ the vector of the line

$$\vec{\mathbf{l}}_k = \frac{\mathbf{s}_k - \mathbf{e}_k}{\|\mathbf{s}_k - \mathbf{e}_k\|_2} \quad (4.5)$$

estimated from the segment endpoints to compare the geometric features of each of them.

Then, given  $\mathcal{L}_1$  and  $\mathcal{L}_2$ , our aim is to find the subset of corresponding line segments between the two input images (see Figure 4.6), defined as  $\mathcal{M}_{12} = \{(l_i, l_j) \mid l_i \in \mathcal{L}_1 \wedge l_j \in \mathcal{L}_2\}$ . For  $l_i$  and  $l_j$  to be a positive match, they must be parallel, have a sufficient overlap and be compliant with the epipolar geometry of the two views. In order to impose the lines to be parallel, we consider the angle formed by the two line segments in the image plane,  $\theta_{ij}$ :

$$\theta_{ij} = \text{atan}(\|\vec{\mathbf{l}}_i \times \vec{\mathbf{l}}_j\| / \|\vec{\mathbf{l}}_i \cdot \vec{\mathbf{l}}_j\|). \quad (4.6)$$

The above-mentioned expressions, however, might lead to inconsistent results as any line in the image could satisfy Equation (4.6) without being related to the query one. Therefore, we deal with this phenomena by also defining the *overlap* of two line segments  $\rho_{ij} \in [0, 1]$  as the ratio between their common parts, as depicted in Figure 4.5, where  $\rho_{ij}$  equals 0 and 1 when there is none or full overlapping between the line segments, respectively. In addition, we also define the ratio between the line lengths as:

$$\mu_{ij} = \frac{\max(L_i, L_j)}{\min(L_i, L_j)} \quad (4.7)$$

where  $L_k = \|\mathbf{s}_k - \mathbf{e}_k\|_2$  stands for the length of the  $k$ -th line, which discards any likely pair of segments whose lengths are not similar enough (if they are of similar length the value of  $\mu_{ij}$  is close to one, and bigger than one otherwise).

Finally, we also consider epipolar geometry as a possible constraint for the two different cases of study. In the first case, *stereo* matching, we define the angle formed by the middle point flow vector,  $\mathbf{x}_{ij} = \mathbf{m}_i - \mathbf{m}_j$  where the middle point is defined as  $\mathbf{m}_k = (\mathbf{s}_k + \mathbf{e}_k)/2$ , as:

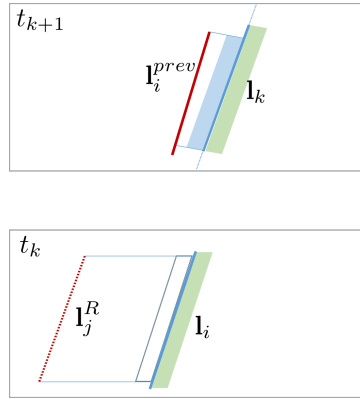
$$\theta_{ij}^{st} = \text{asin}(\|\mathbf{x}_{ij} \times \boldsymbol{\eta}_1\| / \|\mathbf{x}_{ij}\|) \quad (4.8)$$

where  $\boldsymbol{\eta}_1$  stands for the director vector of the  $X$  direction. In contrast, in the *frame-to-frame* case, we assume that images are separated by a small motion and therefore we define the angle formed by  $\mathbf{x}_{ij}$  and the  $Y$  direction (whose unit vector is given by  $\boldsymbol{\eta}_2$ ), namely:

$$\theta_{ij}^{ff} = \text{asin}(\|\mathbf{x}_{ij} \times \boldsymbol{\eta}_2\| / \|\mathbf{x}_{ij}\|). \quad (4.9)$$

### Sparse $\ell_1$ -Minimization for Line Segment Tracking

In this paper, we formulate line-segment tracking as a sparse minimization problem solely based on the previously introduced geometric constraints. Although this representation has been already employed in computer vision for noise reduction [48], face recognition [30], and loop closure detection [93]



**Figure 4.5:** Scheme of the line segment overlap for both the *stereo* and *frame-to-frame* cases. In the bottom image  $t_k$  we plot the stereo overlap between the reference line in the left image  $l_i$  and a match candidate in the right one  $l_j^R$ , which is the ratio of the lengths of the shadowed areas (in blue the overlap and in green the line's length). Similarly, the above image  $t_{k+1}$  depicts the overlap between the reference line in the second image  $l_k$  and the projected line in the second frame  $l_i^{prev}$ .

(among others), to the best of our knowledge this is the first time it is employed for the geometric tracking of line segment features. For that, we also take advantage of the *1-sparse* nature of the tracking problem, i.e. a single line  $l_i$  from the first image should only have at most one match candidate from the  $\mathcal{L}_2$  set. It must be noticed that, in the case of detecting divided lines, it is possible for more than one line to match the query one, however, this case is even more likely to occur with appearance based methods, as any locally similar line in the image can be a candidate.

Let us define the  $n$ -dimensional *matching* vector  $\omega_i$  of the line  $l_i \in \mathcal{L}_1$  as:

$$\omega_i = [\omega_{i0} \dots \omega_{ij} \dots \omega_{in}]^\top \quad (4.10)$$

where  $\omega_{ij}$  equals one if  $l_i$  and  $l_j$  are positive matches and zero otherwise, and  $n$  stands for the number of line segments in  $\mathcal{L}_2$ . Moreover, we define the line segment *error* vectors  $\beta_{ij}$  and the objective  $b$  for both the *stereo* and *frame-to-frame* cases as:

$$\beta_{ij} = \begin{bmatrix} \theta_{ij} \\ \theta_{ij}^{epip} \\ \rho_{ij} \\ \mu_{ij} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad (4.11)$$

for the line segments  $l_i \in \mathcal{L}_1$  and  $l_j \in \mathcal{L}_2$ , where *epip* refers to the epipolar constraints defined in Equations (4.8) and (4.9) for the two cases of study.

Now, by concatenating all line segment error vectors we form the  $4 \times n$  matrix  $\mathbf{A}_i$ :

$$\mathbf{A}_i = [\beta_{i0}, \dots, \beta_{ij}, \dots, \beta_{in}]. \quad (4.12)$$

that must satisfy the linear constraint  $\mathbf{A}_i \boldsymbol{\omega}_i = \mathbf{b}$  if the sum over all the components from the matching vector  $\boldsymbol{\omega}_i$  is one (which is our hypothesis). While  $\ell_2$ -norm is usually employed to solve the previous problem with the typical least-squares formulation, it is worth noticing that it leads to a dense representation of the optimal  $\boldsymbol{\omega}_i^*$ , which contradicts the 1-sparse nature of our solution.

In contrast, we can formulate the problem of finding  $l_j \in \mathcal{L}_2$  that properly matches  $l_i \in \mathcal{L}_1$  as a *convex, sparse, constrained*  $\ell_1$ -minimization as follows:

$$\min_{\boldsymbol{\omega}_i} \|\boldsymbol{\omega}_i\|_1 \text{ subject to } \|\mathbf{A}_i \boldsymbol{\omega}_i - \mathbf{b}\|_2 \leq \epsilon \quad (4.13)$$

where the constraint corresponds to the above-mentioned geometrical conditions, and  $\epsilon > 0$  is the maximum tolerance for the constraint error. Moreover, the problem in Equation (4.13) can be also solved with the homotopy approach [14] in the following *unconstrained* manner:

$$\min_{\boldsymbol{\omega}_i} \lambda \|\boldsymbol{\omega}_i\|_1 + \frac{1}{2} \|\mathbf{A}_i \boldsymbol{\omega}_i - \mathbf{b}\|_2^2 \quad (4.14)$$

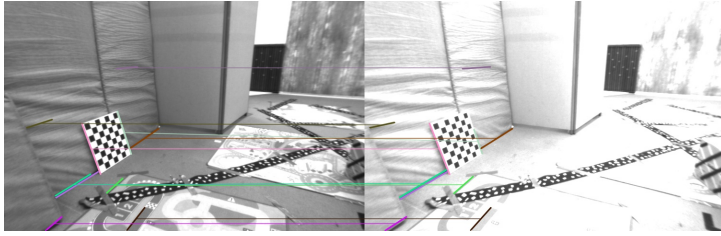
with  $\lambda$  a weighting parameter empirically set to 0.1, resulting in a very effective and fast solver [41].

Then, we efficiently solve the problem in Equation (4.14) for each  $l_i \in \mathcal{L}_1$  obtaining the sparse vector  $\boldsymbol{\omega}_i$ , which after being normalized indicates whether the line segment  $l_i$  has a positive match (in the maximum entry  $j$  of  $\boldsymbol{\omega}_i$ ). Finally, we guarantee that line segments are uniquely corresponded by only considering the candidate with minimum error, defined as  $\|\beta_{ij}\|$ , if the error for the second best match is at least 2 times bigger than the best one. For further details on the mathematics of this Section, please refer to [14].

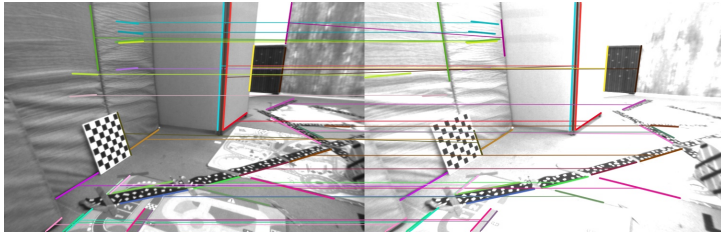
### Dealing with Outliers

When dealing with repetitive structures a number of outliers can appear. To deal with this problem in the *stereo* case, we implement a filter based on the epipolar constraint, for which we first estimate robustly the normal distribution formed by the angles with the horizontal direction. Then, we discard the matches whose angle with the horizontal direction lies above 2 times the standard deviation of the distribution formed by all matches.

In the *frame-to-frame* case, as the camera pose is not known yet, we cannot directly apply epipolar geometry. However, we approximate an epipolar filter, based on the assumption that input images belong to consecutive frames from a sequence, and therefore they are separated by small motions. For that, we



(a) LSD [147] + LBD [154]



(b) LSD [147] + Our proposal

**Figure 4.6:** Stereo correspondences between two different images from the EuRoC dataset. Our matching algorithm is capable of finding matches that does not necessarily have similar appearance.

discard the matches whose angle with the vertical direction (this is the epipolar constraint in the case of null motion) lies above 2 times the standard deviation of the distribution formed by all matches as they are less likely to fulfil the motion constraints.

#### 4.D.4 Point-Segment Visual Odometry Overview

In this section we briefly describe the PLVO stereo visual odometry system [62] where the proposed matching algorithm has been integrated for line segment tracking. PLVO combines probabilistically both point and line segment features and its C++ implementation is available publicly <https://github.com/rubengooj/StV0-PL>.

##### 4.D.4.1 Point Features

In PLVO points are detected and described with ORB [126] (consisting of a FAST keypoint detector and a BRIEF descriptor) due to its efficiency and good performance. In order to reduce the number of outliers, we only consider the measurements that are mutual best matches, and also check that the two best matches are significantly separated in the description space by only accepting

matches whose distance between the two closest correspondences is above the double of the distance to the best match.

#### 4.D.4.2 Line Segment Features

In our previous work [62] we detect line segments with the Line Segment Detector (LSD) [147] and also employ the Line Band Descriptor (LBD) [154] for the stereo and frame-to-frame matching. Although this method provides a high precision and repeatability, it still presents very high computational requirements, simple detection and matching requires more than 30ms with  $752 \times 480$ , thus its use limits their application in real-time. In order to reduce the computational burden of the Stereo VO system, in this work we have also employed the Fast Line Detector (FLD) [94], which is based on connecting collinear Canny edges [27]. This detector works faster than LSD at expense of a poorer performance in detecting meaningful lines, i.e. lines with strong local support along all the lines, since, unlike LSD, detection is only based on image edges.

#### 4.D.4.3 Motion Estimation

After obtaining a set of point and line correspondences, we then recover the camera motion with iterative Gauss-Newton minimization of the projection errors in each case (in the case of line segments we employ the distance from the projected endpoint, to the line in the next frame). To mitigate the undesirable effect of outliers and noisy measurements, we perform a two steps minimization for which we weight the observations with a Pseudo-Huber loss function, and then we remove the outliers and refine the solution.

### 4.D.5 Experimental Validation

We evaluate the performance and robustness of our proposal in several public datasets for two different line segment detectors, LSD [147] and FLD [94], when employing two different matching strategies: our proposal, and traditional appearance based tracking with LBD [154]. All the experiments have run on an Intel Core i7-3770 CPU @ 3.40 GHz and 8GB RAM without GPU parallelization. In our experiments we have employed a fixed number of detected lines set to 100, and 600 ORB [126] features for the case of points and line based VO.

#### Tracking Performance

First, we compare the line segment tracking performance of our proposal against traditional feature matching approaches. For that, we took several sequences (at different speeds) and classified each match as an inlier if the

**Table 4.4:** Tracking performance of our proposal and traditional line segment feature matching (number of matches - inliers).

Dataset	Resolution	LSD + LBD		LSD + L1		FLD + LBD		FLD + L1	
hdr/bear	640 × 480	72	80 %	39	94 %	65	73 %	45	86 %
hdr/desk	640 × 480	70	84 %	40	93 %	60	83 %	41	88 %
hdr/floor1	640 × 480	27	77 %	22	85 %	29	77 %	33	85 %
hdr/floor2	640 × 480	71	80 %	42	93 %	61	80 %	41	87 %
hdr/sofa	640 × 480	58	81 %	41	90 %	49	81 %	45	84 %
hdr/whiteboard	640 × 480	41	76 %	35	95 %	42	75 %	35	90 %
hdr/flicker1	640 × 480	42	84 %	45	98 %	44	80 %	40	95 %
hdr/flicker2	640 × 480	86	89 %	45	98 %	74	85 %	45	97 %
dnn/1-light	752 × 480	16	94 %	15	100 %	19	90 %	18	100 %
dnn/2-lights	752 × 480	29	94 %	18	100 %	38	92 %	24	97 %
dnn/3-lights	752 × 480	49	90 %	35	100 %	57	91 %	35	97 %
dnn/change-light	752 × 480	19	85 %	14	100 %	24	95 %	18	100 %
dnn/hdr1	752 × 480	55	90 %	35	100 %	51	88 %	35	95 %
dnn/hdr2	752 × 480	50	90 %	32	97 %	52	93 %	39	95 %
dnn/overexp	752 × 480	84	94 %	55	98 %	75	92 %	47	96 %
dnn/overexp-change-light	752 × 480	82	92 %	50	98 %	72	90 %	43	96 %
dnn/low-texture	752 × 480	53	88 %	35	100 %	52	90 %	35	97 %
dnn/low-texture-rot	752 × 480	43	90 %	30	100 %	40	90 %	30	95 %
tsukuba	640 × 480	43	86 %	34	84 %	36	75 %	21	86 %
tsukuba/fluor(L)-daylight(R)	640 × 480	5	40 %	15	80 %	5	40 %	12	80 %
tsukuba/fluor(L)-flashlight(R)	640 × 480	2	33 %	5	60 %	2	50 %	4	50 %
tsukuba/fluor(L)-lamps(R)	640 × 480	1	0 %	5	60 %	1	0 %	3	67 %

correspondent line segment projection error is less than one pixel when employing the groundtruth transformation. In order to compare the algorithms under dynamic illumination changes, we have employed two specific datasets: one extracted from [96] (*hdr*) taken with an RGB-D sensor under HDR situations, and another one from our previous work [66] (*dnn*) containing a number of difficult dynamic illumination conditions. In addition, we also have employed the Tsukuba Stereo Dataset [120], a synthetic dataset rendered under 4 different illuminations, i.e. *fluorescent*, *lamps*, *flashlight*, and *daylight*. For a more challenging set of experiments, we have also employed to use all combinations (taking *fluorescent* as reference) of the rendered sequences, by setting the left one to the reference and the right one to all different possibilities. It is worth noticing that illumination changes from the considered datasets are produced punctually, and after that, the scene illumination usually keeps constant until the next change. This benefits to descriptor-based techniques when evaluating the tracking performance during the whole sequence, for which we also recommend to watch the attached video for visual evaluation under such circumstances.

Table 4.4 shows the tracking accuracy and the number of features tracked, for all the sequences from each considered dataset. First, we observe a slightly inferior performance of FLD [94] in comparison against LSD [147], due to its lower repeatability in contrast with its superior computational performance

**Table 4.5:** Relative RMSE errors in the EuRoC MAV dataset [23].

Sequence	LVO (FLD)	LVO-L1 (FLD)	LVO (LSD)	LVO-L1 (LSD)
MH-01-easy	0.0641	0.0788	0.0669	0.0716
MH-02-easy	0.0826	0.0923	0.0740	0.0881
MH-03-med	0.0886	0.1011	0.0898	0.1004
MH-04-diff	0.1500	0.1536	0.1429	0.1518
MH-05-diff	0.1350	0.1529	0.1391	0.1561
V1-01-easy	0.0890	0.0969	0.0876	0.0954
V1-02-med	0.0662	0.0847	0.0606	0.0947
V1-03-diff	0.2261	0.1518	0.0765	0.1103
V2-01-easy	0.1980	0.1868	0.1662	0.1898
V2-02-med	0.1634	0.2294	0.1982	0.2562
V2-03-diff	0.2329	0.2342	0.2354	0.2275
MH-01-easy*	0.0787	0.0897	0.0741	0.0728
MH-02-easy*	0.0873	0.1015	0.8237	0.0981
MH-03-med*	0.0982	0.1578	0.0916	0.1141
MH-04-diff*	0.1540	0.1780	0.1354	0.1621
MH-05-diff*	-	0.1603	-	0.1863
V1-01-easy*	0.0880	0.1011	0.0997	0.1041
V1-02-med*	0.0858	0.0953	0.0713	0.1096
V1-03-diff*	-	0.2087	-	0.1598
V2-01-easy*	-	0.2396	-	0.2080
V2-02-med*	-	0.2472	-	0.2563
V2-03-diff*	-	-	-	0.2631

(see Table 4.6). In general, we observe that our matching method decreases the number of features, due to the very requiring assumptions of our matching technique, however, it provides a higher ratio of inliers thanks to the extra stage explained in Section 4.D.3.

As for the Tsukuba dataset, we observe that the number of features successfully tracked dramatically decreases as the response of the detectors is not capable of producing a compatible set of lines from the same images. However, we observe that our method technique is capable of recovering more matches, specially in the less challenging case (*fluorescent* and *daylight*), that can be employed along different sensing to extract more information from the environment in such difficult situations.

### Robustness Evaluation in Stereo Visual Odometry

In this set of experiments, we test the performance of the compared algorithms in the EuRoC [23] dataset. In order to simulate changes in exposure time or il-

**Table 4.6:** Comparison of the computational performance of the different considered algorithms.

	Monocular Tracking	Stereo Tracking
LSD + LBD	39.342 ms	51.347 ms
LSD + Our	25.897 ms	35.828 ms
FLD + LBD	18.147 ms	33.266 ms
FLD + Our	7.654 ms	23.445 ms

lumination within the EuRoC dataset [23] (we will refer to simulated sequences with an asterisk) we change the gain and bias of the image with two uniform distribution, i.e.  $\alpha = \mathcal{U}(0.5, 2.5)$  and  $\beta = \mathcal{U}(0, 20)$  pixels every 30 seconds. For that comparison, we not only focus in the accuracy of the estimated trajectories, but also in the robustness of the algorithms under different environment conditions (we mark a dash those experiments where the algorithm lose the track). We compare the accuracy of trajectories obtained with our previous stereo VO system, PLVO [62], against our proposal tracking strategy, PLVO-L1, when employing LSD or FLD features. Table 4.5 contains the results by computing the relative RMSE in translation for the estimated trajectories. As we can observe, in the raw dataset our approach performs slightly worse than standard appearance-based tracking techniques, mainly due to the lower number of correspondences provided by our algorithm, as mentioned in previous Section. In contrast, we can observe a considerable decrease in accuracy of our approaches, however, they are capable of estimating the motion in all sequences with an lower accuracy, mainly due to the less number of matches, due to restrictive constraints. For this reason we believe our matching technique a suitable option to address the line segment tracking problem under severe appearance changes, in combination with prior information from different sensors and/or algorithms.

### Computational Cost

Finally, we compare the computational performance of the different tracking algorithms in the considered datasets considering the time of processing one image (similarly to the VO framework). In the both cases we can observe the superior performance of our proposal, it runs between 1.5 and 2 times faster depending on the detector employed, thanks to the efficient implementation of the geometric-based tracking thus making it very suitable for robust real-time application, most likely in combination with other sensing, such as inertial measurement unit sensors (IMU).

### 4.D.6 Conclusions

In this work, we have proposed a geometrical approach for the robust matching of line segments for challenging stereo streams, such as sequences including severe illumination changes or HDR environments. For that, we exploit the nature of the matching problem, i.e. every observation can only be corresponded with a single feature in the second image or not corresponded at all, and hence we state the problem as a sparse, convex,  $\ell_1$ -minimization of the matching vector regularized by the geometric constraints. Thanks to this formulation we are able of robustly tracking line segments along sequences recorded under dynamic changes in illumination conditions or in HDR scenarios where usual appearance-based matching techniques fail. We validate the claimed features by first evaluating the matching performance in challenging video sequences, and then testing the system in a benchmarked point and line based VO algorithm showing promising results.

## Conclusions

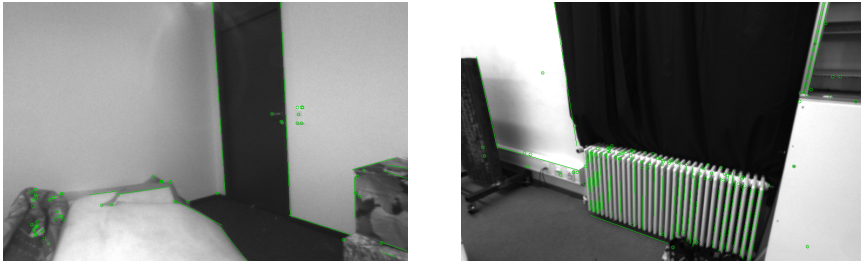
In the last 20 years, visual SLAM techniques have reached remarkable maturity with impressive results achieved in controlled environments. In fact, SLAM has been considered a theoretically solved problem for the past decade, as stated by Durrant-Whyte and Bailey in 2006 [43]: *At a theoretical and conceptual level, SLAM can now be considered a solved problem. However, substantial issues remain in practically realizing more general SLAM solutions and notably in building and using perceptually rich maps as part of a SLAM algorithm.*

Today, more than a decade later, it is still one of the most active research topics in computer vision and mobile robotics, and the question of *Is SLAM solved?* is often asked for the scientific community [24]. One of the reasons behind is that, despite the maturity reached by state-of-art visual SLAM techniques in controlled environments, there are still many open challenges to address before reaching a SLAM system robust to long-term operations in uncontrolled scenarios, where classical assumptions, such as static environments, do not hold.

This thesis has contributed to overcome some of the aforementioned limitations of traditional visual SLAM and/or odometry techniques by addressing the problem from different perspectives. Specifically, this work aims to advance towards a robust visual SLAM system that mitigates the limitation of current techniques, *i.e.*, robustness to different types of environment, challenging illuminations, etc. In this context, the scope of this thesis comprehends on one hand the design and implementation of new perception and navigation algorithms that provide accurate location and some type of representation of the environment, and, on the other, the integration of such approaches along with technologies in real world applications, such as mobile robotics. The main contributions of this thesis can be grouped into two major topics described in the following parts.

### Contributions to SLAM in Low-textured Environments

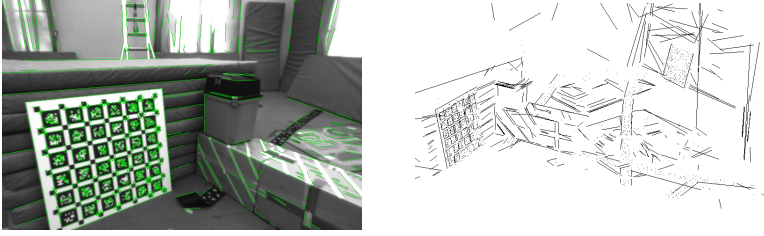
The first set of works focuses on improving the robustness of visual odometry and SLAM techniques in low-textured environments (see Figure 5.1), where it is common that the performance of traditional approaches decreases due to difficulties in a sufficient number of reliable point features. In such cases, the effect is an accuracy impoverishment and, occasionally, the absolute failure of the system.



**Figure 5.1:** Low-textured scenes are challenging for typical VO/SLAM systems.

This group of works benefits from an alternative feature choice, *i.e.*, line segments, to exploit the information from the structured parts of the environment. For that, we have contributed with:

- *Robust Stereo Visual Odometry through a Probabilistic Combination of Points and Line Segments* [62]. In this work, we implemented a stereo visual odometry framework to effectively combine points and line segments, achieving a good performance in both structured and low-textured scenarios. Despite being one of our earliest contributions, we managed to achieve a versatile system, capable of leveraging the impact of the different features based on their uncertainty, that worked closely in real-time (between 10-30 Hz depending on the resolution).
- *Accurate Stereo Visual Odometry with Gamma Distributions* [123]. Shortly after the previous contribution, we focused on achieving a more accurate modeling of the errors, by using a Gamma distribution over the residual magnitudes, rather than a Gaussian over the projection errors. We demonstrated, both in simulation and real data experiments, that using this error models along our previous odometry system allowed us to achieve more accurate performance.
- *PL-SVO: Semi-Direct Monocular Visual Odometry by Combining Points and Line Segments* [61]. On the other hand, dealing with line segment features in images is not as straightforward as the case of point features, since



**Figure 5.2:** Mapping results obtained with PL-SLAM in a sequence from the EuRoC MAV dataset.

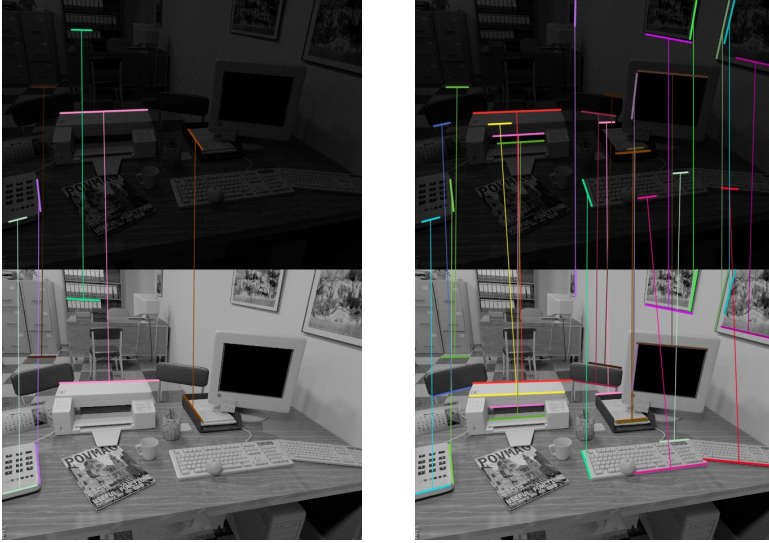
they are difficult to represent while also requiring higher computational burden for its tracking. Indeed, this was one of the main bottlenecks of [62]. To alleviate this additional difficulties, we benefited from the semi-direct approach to monocular odometry to extend a previous state-of-art work, known as SVO [53]. This allowed for a faster feature tracking, performing at almost 60 Hz in public datasets, since the semi-direct framework eliminated the necessity of a continuous feature detection and matching.

- *PL-SLAM: a Stereo SLAM System through the Combination of Points and Line Segments* [65]. Finally, we decided to extend , while also improving its implementation, our previous stereo system to a complete real-time stereo SLAM system. For that, we leveraged the importance of both types of features at all instances of the process: visual odometry, keyframe selection, bundle adjustment, etc. Among the benefits, the resulting system is more robust to difficult environments and, additionally, the estimated maps are richer (see Figure 5.2) and more diverse in 3D elements, which can be exploited to infer valuable, high-level scene structures like planes, empty spaces, ground plane, etc.

### Contributions to SLAM under Challenging Illumination

One of the main open challenges in visual odometry and SLAM is its *robustness* to difficult *illumination* conditions or *high dynamic range (HDR)* environments (see Figure 5.3). In such cases, difficulties come from both the limitations of the sensors, *e.g.* , quick changes from dark to bright areas might over-expose the images, and the inability to perform a successful tracking of interest points because of the bold assumptions in SLAM such as brightness constancy. The work of this thesis contributes to these phenomena from two different perspectives.

- *Learning-based Image Enhancement for Visual Odometry in Challenging HDR Environments* [66]. Initially, we addressed this problem with a deep



**Figure 5.3:** Tracking in environments with changes in illumination conditions or in HDR scenarios is a challenge for traditional appearance-based matching techniques.

learning approach by enhancing monocular images to more informative and invariant representations for VO and SLAM, since deep neural networks have proved to achieve robust performance in varied conditions thanks to their generalization properties. This work have also demonstrated how the insertion of long short term memory (LSTM) networks allowed for temporally consistent sequences since the estimation was also depending on previous states. As a proof of concept, we then compared the performance of two of the state-of-art algorithms for monocular VO, showing the benefits of using the enhanced images, by achieving a more stable and accurate performance in challenging environments.

- *Geometric-based Line Segment Tracking for HDR Stereo Sequences [63]*. Our second approach adopted a more traditional perspective contributing with a purely geometrical approach for the *robust matching* of line segments for challenging stereo streams with severe illumination changes or High Dynamic Range (HDR) environments. In this contribution we proved that line segments can be successfully tracked along video sequences by only considering their geometric consistency, and validates it by evaluating both the matching performance and motion estimation in challenging video sequences from benchmark datasets. This allowed us to achieve good tracking in extremely difficult situations while also accelerating the tracking 1.5-2 times than the reference one.

## Future Work

Apart from improvements in robustness, which motivated this thesis work, there are many other interesting topics to achieve a robust SLAM system capable of working in arbitrary scenarios:

*Long Term Appearance-based SLAM.* A different approach to the visual SLAM problem to the ones summarized in Chapter 2, which in turn estimates the relative pose with respect to the map, is based on appearance. One of the benefits from such approach is its robustness to drastic visual changes, such as the ones produced between sequences taking during the day and night, seasonal changes, or long-term structural changes. On the contrary, those methods are not currently suitable for *metric relocation*, where feature-based approaches are still the only possibility, however, they do not provide invariance to such dramatic appearance changes.

*Active SLAM.* A recently emerging research line that, in general, tries to employ the incoming information in order to predict automatically the optimal settings for each situation. For instance, most SLAM implementations do require extensive parameter tuning which typically is empirically set for a given scenario, and this may not suffice in arbitrary scenarios where *automatic parameter adjustment* techniques might highly benefit the algorithms. Other examples can be the control of the camera parameters with predicted values to maximize for instance the visual information, or provide robots with mobile cameras able to predict which part of the map is more informative for the assigned task.

*Semantic Maps.* Typical SLAM methods consist of a set of 3D landmarks which can be used for robotics tasks such as obstacle avoidance or navigation, but its main advantage is the reduction of the modeled errors for a more accurate localization. On the other hand, this type of maps are highly limited to perform, for instance, more complex tasks such as object/person recognition, or higher level robotic missions, *e.g.* "Go to the kitchen", where *semantic* knowledge of the environment is required. To illustrate this, an autonomous car SLAM application could benefit from the use of semantic information from the surroundings to predict the robot pose from the static objects while at the same time estimating the state of the dynamic objects.

*Deep-learning in SLAM.* Finally, another group of techniques is starting to emerge in the SLAM community, *i.e.*, those employing *deep-learning* approaches to provide the systems with high-level knowledge, hardly to achieve with purely geometric techniques [36]. For instance, in [105] authors combine CNNs with geometric SLAM, to provide semantically labeled 3D maps in real-time. Deep learning has also been used to improve traditional SLAM techniques, *e.g.* [21], or even to propose a keyframe-based dense camera tracking and depth map estimation that is entirely learned [156].



## Bibliography

- [1] Fotokite. <https://fotokite.com/>.
- [2] Google Earth VR. <https://vr.google.com/earth/>.
- [3] Google Self-Driving Car Project. <https://www.google.com/selfdrivingcar/>.
- [4] Nuro AI. <https://nuro.ai/>.
- [5] Oculus VR. <https://www.oculus.com/>.
- [6] Robot Dyson 360 Eye. <http://www.dyson360eye.com/>.
- [7] Tesla. <https://www.tesla.com/>.
- [8] Waymo. <https://waymo.com/>.
- [9] G. Agamennoni, P. Furgale, and R. Siegwart. Self-tuning M-estimators. pages 4628–4635, 2015.
- [10] G. Agamennoni, J. I. Nieto, and E. M. Nebot. An outlier-robust Kalman filter. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1551–1558. IEEE, 2011.
- [11] G. Agamennoni, J. I. Nieto, and E. M. Nebot. Approximate inference in state-space models with heavy-tailed noise. *IEEE Transactions on Signal Processing*, 60(10):5024–5037, 2012.

- [12] P. F. Alcantarilla, J. J. Yebes, J. Almazán, and L. M. Bergasa. On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1290–1297. IEEE, 2012.
- [13] H. Alismail, M. Kaess, B. Browning, and S. Lucey. Direct Visual Odometry in Low Light using Binary Descriptors. *IEEE Robotics and Automation Letters*, 2016.
- [14] M. S. Asif. *Primal Dual Pursuit: A homotopy based algorithm for the Dantzig selector*. PhD thesis, Georgia Institute of Technology, 2008.
- [15] G. J. Babu and C. Rao. Goodness-of-fit tests when parameters are estimated. *Sankhyā: The Indian Journal of Statistics*, pages 63–74, 2004.
- [16] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.
- [17] S. Baker and I. Matthews. Lucas-Kanade 20 Years On : A Unifying Framework : Part 1 2 Background : Lucas-Kanade. *International Journal of Computer Vision*, 56(3):221–255, 2004.
- [18] A. Bartoli and P. Sturm. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. *Computer Vision and Image Understanding*, 100(3):416–441, 2005.
- [19] J.-l. Blanco. A tutorial on SE (3) transformation parameterizations and on-manifold optimization. (3), 2013.
- [20] J.-L. Blanco-Claraco, F.-Á. Moreno-Dueñas, and J. González-Jiménez. The Málaga urban dataset: High-rate stereo and LiDAR in a realistic urban scenario. *The International Journal of Robotics Research*, 33(2):207–214, 2014.
- [21] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison. CodeSLAM—learning a compact, optimisable representation for dense visual SLAM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2560–2568, 2018.
- [22] J. Briales and J. Gonzalez-Jimenez. A Minimal Closed-form Solution for the Perspective Three Orthogonal Angles (P3oA) Problem: Application To Visual Odometry. *Journal of Mathematical Imaging and Vision*, 55(3):266–283, 2016.
- [23] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart. The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

- [24] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.
- [25] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. D. Reid, and J. J. Leonard. Simultaneous Localization And Mapping: Present, Future, and the Robust-Perception Age. *CoRR*, abs/1606.05830, 2016.
- [26] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.
- [27] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.
- [28] D. Caruso, J. Engel, and D. Cremers. Large-scale direct slam for omnidirectional cameras. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 141–148. IEEE, 2015.
- [29] M. Chandraker, J. Lim, and D. Kriegman. Moving in stereo: Efficient structure and motion using lines. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1741–1748. IEEE, 2009.
- [30] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. S. Huang. Learning with  $\ell^1$ -graph for image analysis. *IEEE transactions on image processing*, 19(4):858–866, 2010.
- [31] J. Civera, A. J. Davison, and J. M. Montiel. Inverse depth parametrization for monocular SLAM. *IEEE transactions on robotics*, 24(5):932–945, 2008.
- [32] B. R. Clarke, P. L. McKinnon, and G. Riley. A fast robust method for fitting gamma distributions. *Statistical Papers*, 53(4):1001–1014, 2011.
- [33] J. C. Clarke. Modelling uncertainty: A primer. *Tutorial of Department of Eng. Science*, pages 1–21, 1998.
- [34] J. M. Coughlan and A. L. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 941–947. IEEE, 1999.
- [35] M. Cummins and P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

- [36] A. J. Davison. FutureMapping: The computational structure of spatial AI systems. *arXiv preprint arXiv:1803.11288*, 2018.
- [37] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):1052–1067, 2007.
- [38] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH 2008 classes*, page 31. ACM, 2008.
- [39] F. Dellaert, M. Kaess, et al. Factor graphs for robot perception. *Foundations and Trends® in Robotics*, 6(1-2):1–139, 2017.
- [40] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, pages 1009–1014. IEEE, 2000.
- [41] D. L. Donoho and Y. Tsaig. Fast solution of  $\ell^1$ -norm minimization problems when the solution may be sparse. *IEEE Transactions on Information Theory*, 54(11):4789–4812, 2008.
- [42] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. FlowNet: Learning optical flow with convolutional networks. In *ICCV 2015*, pages 2758–2766, 2015.
- [43] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.
- [44] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (SLAM). *IEEE Robotics & Automation Magazine*, 13(2):99–116, 2006.
- [45] E. Eade and T. Drummond. Edge Landmarks in Monocular SLAM. In *BMVC*, pages 7–16, 2006.
- [46] E. Eade and T. Drummond. Unified Loop Closing and Recovery for Real Time Monocular SLAM. In *BMVC*, volume 13, page 136. Citeseer, 2008.
- [47] E. Eade and T. Drummond. Edge landmarks in monocular SLAM. *Image and Vision Computing*, 27(5):588–596, apr 2009.
- [48] M. Elad, M. A. Figueiredo, and Y. Ma. On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE*, 98(6):972–982, 2010.

- [49] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2018.
- [50] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.
- [51] J. Engel, J. Sturm, and D. Cremers. Semi-Dense Visual Odometry for a Monocular Camera. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1449–1456. IEEE, 2013.
- [52] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation. In *Robotics: Science and Systems XI*, number EPFL-CONF-214687, 2015.
- [53] C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 15–22. IEEE, 2014.
- [54] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. SVO: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265, 2017.
- [55] F. Fraundorfer and D. Scaramuzza. Visual odometry: Part II: Matching, robustness, optimization, and applications. *IEEE Robotics & Automation Magazine*, 19(2):78–90, 2012.
- [56] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, et al. Event-based vision: A survey. *arXiv preprint arXiv:1904.08405*, 2019.
- [57] D. Gálvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.
- [58] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.
- [59] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 963–968. IEEE, 2011.
- [60] A. Gelb. *Applied optimal estimation*. MIT press, 1974.

- [61] R. Gomez-Ojeda, J. Briales, and J. González-Jiménez. PL-SVO: Semi-Direct Monocular Visual Odometry by Combining Points and Line Segments. In *Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4211–4216. IEEE/RSJ, 2016.
- [62] R. Gomez-Ojeda and J. Gonzalez-Jimenez. Robust stereo visual odometry through a probabilistic combination of points and line segments. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2521–2526, May 2016.
- [63] R. Gomez-Ojeda and J. Gonzalez-Jimenez. Geometric-based Line Segment Tracking for HDR Stereo Sequences. In *Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE/RSJ, 2018.
- [64] R. Gomez-Ojeda, M. Lopez-Antequera, N. Petkov, and J. Gonzalez-Jimenez. Training a convolutional neural network for appearance-invariant place recognition. *arXiv preprint arXiv:1505.07428*, 2015.
- [65] R. Gomez-Ojeda, F.-A. Moreno, D. Zuñiga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez. PL-SLAM: a stereo SLAM system through the combination of points and line segments. *IEEE Transactions on Robotics*, 35(3):734–746, 2019.
- [66] R. Gomez-Ojeda, Z. Zhang, J. Gonzalez-Jimenez, and D. Scaramuzza. Learning-based image enhancement for visual odometry in challenging HDR environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 805–811. IEEE, 2018.
- [67] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Computational Intelligence in Robotics and Automation, 1999. CIRA '99. Proceedings. 1999 IEEE International Symposium on*, pages 318–325. IEEE, 1999.
- [68] 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.
- [69] C. Harris and C. Stennett. RAPID-a video rate object tracker. In *BMVC*, pages 1–6, 1990.
- [70] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [71] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

- [72] M. Hofer, M. Maurer, and H. Bischof. Line3D: Efficient 3D Scene Abstraction for the Built Environment. In *German Conference on Pattern Recognition*, pages 237–248. Springer, 2015.
- [73] P. J. Huber. *Robust statistics*. Springer, 2011.
- [74] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [75] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.
- [76] M. Jaimez and J. Gonzalez-Jimenez. Fast visual odometry for 3-D range sensors. *Robotics, IEEE Transactions on*, 31(4):809–822, 2015.
- [77] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proc. of the 22nd ACM Int. Conf. on Multimedia*, pages 675–678. ACM, 2014.
- [78] H. Jin, P. Favaro, and S. Soatto. Real-time 3D motion and structure of point features: a front-end system for vision-based control and interaction. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 2, pages 778–779. IEEE, 2000.
- [79] H. Jin, P. Favaro, and S. Soatto. Real-time feature tracking and outlier rejection with changes in illumination. In *ICCV 2001*, volume 1, pages 684–689. IEEE, 2001.
- [80] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008.
- [81] A. Kendall, V. Badrinarayanan, , and R. Cipolla. Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding. *arXiv preprint arXiv:1511.02680*, 2015.
- [82] C. Kerl. *Odometry from rgb-d cameras for autonomous quadcopters*. PhD thesis, Citeseer, 2012.
- [83] C. Kerl, J. Sturm, and D. Cremers. Dense visual SLAM for RGB-D cameras. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2100–2106. IEEE, 2013.

- [84] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for rgb-d cameras. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3748–3754. IEEE, 2013.
- [85] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [86] B. Kitt, A. Geiger, and H. Lategahn. Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 486–492. IEEE, 2010.
- [87] G. Klein and D. Murray. Improving the agility of keyframe-based SLAM. In *Computer Vision–ECCV 2008*, pages 802–815. Springer, 2008.
- [88] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.
- [89] G. Klein and D. Murray. Parallel Tracking and Mapping on a Camera Phone. In *Proc. Eighth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'09)*, pages 83–86, Orlando, October 2009.
- [90] T. Koletschka, L. Puig, and K. Daniilidis. MEVO: Multi-environment stereo visual odometry. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 4981–4988. IEEE, 2014.
- [91] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3607–3613. IEEE, 2011.
- [92] M. Kuse and S. Shen. Robust camera motion estimation using direct edge alignment and sub-gradient method. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 573–579, May 2016.
- [93] Y. Latif, G. Huang, J. Leonard, and J. Neira. Sparse optimization for robust and efficient loop closing. *Robotics and Autonomous Systems*, 93:13–26, 2017.
- [94] J. H. Lee, S. Lee, G. Zhang, J. Lim, W. K. Chung, and I. H. Suh. Outdoor place recognition in urban environments using straight lines. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5550–5557. IEEE, 2014.

- [95] S. Leutenegger, M. Chli, and R. Siegwart. BRISK: Binary Robust invariant scalable keypoints. pages 2548–2555, Nov. 2011.
- [96] S. Li, A. Handa, Y. Zhang, and A. Calway. HDRFusion: HDR SLAM using a low-cost auto-exposure RGB-D sensor. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 314–322. IEEE, 2016.
- [97] M. Lopez-Antequera, R. Gomez-Ojeda, N. Petkov, and J. Gonzalez-Jimenez. Appearance-invariant place recognition by discriminatively training a Convolutional Neural Network. *Pattern Recognition Letters*, 2017.
- [98] D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (5):441–450, 1991.
- [99] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4(4):333–349, 1997.
- [100] H. Lu, H. Zhang, S. Yang, and Z. Zheng. Camera parameters auto-adjusting technique for robust robot vision. In *ICRA 2010*, pages 1518–1523. IEEE, 2010.
- [101] M. Luperto, J. Monroy, F.-A. Moreno, J. R. Ruiz-Sarmiento, N. Basilico, J. Gonzalez-Jimenez, and N. A. Borghese. A Multi-Actor Framework Centered around an Assistive Mobile Robot for Elderly People Living Alone. In *IEEE International Conference on Intelligent Robots - Workshop on Robots for Assisted Living (IROS)*, 2018.
- [102] L. Ma, C. Kerl, J. Stückler, and D. Cremers. CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM. In *ICRA 2016*, pages 1285–1291. IEEE, 2016.
- [103] M. Mancini, G. Costante, P. Valigi, and T. A. Ciarfuglia. Fast robust monocular depth estimation for Obstacle Detection with fully convolutional networks. In *IROS 2016*, pages 4296–4303. IEEE, 2016.
- [104] M. Mancini, G. Costante, P. Valigi, T. A. Ciarfuglia, J. Delmerico, and D. Scaramuzza. Towards Domain Independence for Learning-Based Monocular Depth Estimation. *IEEE Robotics and Automation Letters*, 2017.
- [105] J. McCormac, A. Handa, A. Davison, and S. Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and automation (ICRA)*, pages 4628–4635. IEEE, 2017.

- [106] C. Mei and E. Malis. Fast central catadioptric line extraction, estimation, tracking and structure from motion. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 4774–4779. IEEE, 2006.
- [107] M. Milford. Vision-based place recognition: how low can you go? *The International Journal of Robotics Research*, 32(7):766–789, 2013.
- [108] M. J. Milford and G. F. Wyeth. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.
- [109] M. J. Milford, G. F. Wyeth, and D. Prasser. RatSLAM: A Hippocampal Model for Simultaneous Localization and Mapping . *Proceeding of the 2004 IEEE international Conference on Robotics & Automation*, pages 403–408, 2004.
- [110] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, et al. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Aaai/iaai*, pages 593–598, 2002.
- [111] F. A. Moreno, J. L. Blanco, and J. Gonzalez. Stereo vision specific models for particle filter-based {SLAM}. *Robotics and Autonomous Systems*, 57(9):955–970, 2009.
- [112] F.-A. Moreno, J.-L. Blanco, and J. González-Jiménez. ERODE: An efficient and robust outlier detector and its application to stereovisual odometry. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4691–4697. IEEE, 2013.
- [113] F.-A. Moreno, J.-L. Blanco, and J. Gonzalez-Jimenez. A constant-time SLAM back-end in the continuum between global mapping and submapping: application to visual stereo SLAM. *The International Journal of Robotics Research*, 35(9):1036–1056, 2016.
- [114] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza. Continuous-Time Visual-Inertial Odometry for Event Cameras. *IEEE Transactions on Robotics*, (99):1–16, 2018.
- [115] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.
- [116] R. Mur-Artal and J. D. Tardós. ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

- [117] A. J. Neves, B. Cunha, A. J. Pinho, and I. Pinheiro. Autonomous configuration of parameters in robotic digital cameras. In *Iberian Conf. on Pattern Recognition and Image Analysis*, pages 80–87. Springer, 2009.
- [118] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2320–2327. IEEE, 2011.
- [119] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20, 2006.
- [120] M. Peris, S. Martull, A. Maki, Y. Ohkawa, and K. Fukui. Towards a simulation driven stereo vision system. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 1038–1042. IEEE, 2012.
- [121] M. Persson, T. Piccini, M. Felsberg, and R. Mester. Robust Stereo Visual Odometry from Monocular Techniques. *IEEE Intelligent Vehicles Symposium (IV), 2015 IEEE*, (IV):686–691, 2015.
- [122] M. Pizzoli, C. Forster, and D. Scaramuzza. REMODE: Probabilistic, monocular dense reconstruction in real time. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2609–2616. IEEE, 2014.
- [123] R. PL-Ojeda, F.-A. Moreno, and J. Gonzalez-Jimenez. Accurate stereo visual odometry with gamma distributions. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 1423–1428. IEEE, 2017.
- [124] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer. PL-SLAM: Real-time monocular visual SLAM with points and lines. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 4503–4508. IEEE, 2017.
- [125] G. Reitmayr and T. Drummond. Going out: robust model-based tracking for outdoor augmented reality. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 109–118. IEEE Computer Society, 2006.
- [126] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.
- [127] J. C. Russ, J. R. Matey, A. J. Mallinckrodt, S. McKay, et al. The image processing handbook. *Computers in Physics*, 8(2):177–178, 1994.
- [128] D. Scaramuzza and F. Fraundorfer. Visual odometry [tutorial]. *Robotics & Automation Magazine, IEEE*, 18(4):80–92, 2011.

- [129] I. Shim, J.-Y. Lee, and I. S. Kweon. Auto-adjusting camera exposure for outdoor robotics using gradient information. In *IROS 2014*, pages 1011–1017. IEEE, 2014.
- [130] D. Sibley, C. Mei, I. D. Reid, and P. Newman. Adaptive relative bundle adjustment. In *Robotics: science and systems*, volume 32, page 33, 2009.
- [131] G. Sibley, C. Mei, I. Reid, and P. Newman. Vast-scale outdoor navigation using adaptive relative bundle adjustment. *The International Journal of Robotics Research*, 29(8):958–980, 2010.
- [132] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [133] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman. The new college vision and laser data set. *The International Journal of Robotics Research*, 28(5):595–599, May 2009.
- [134] P. Smith, I. Reid, and a. J. Davison. Real-Time Monocular SLAM with Straight Lines. *Proceedings of the British Machine Vision Conference 2006*, pages 3.1–3.10, 2006.
- [135] J. Solà, T. Vidal-Calleja, J. Civera, and J. M. M. Montiel. Impact of Landmark Parametrization on Monocular EKF-SLAM with Points and Lines. *International Journal of Computer Vision*, 97(3):339–368, sep 2011.
- [136] J. Solà, T. Vidal-Calleja, and M. Devy. Undelayed initialization of line segments in monocular SLAM. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pages 1553–1558, 2009.
- [137] S. Song and M. Chandraker. Robust scale estimation in real-time monocular SFM for autonomous driving. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1566–1573. IEEE, 2014.
- [138] M. A. Stephens. EDF statistics for goodness of fit and some comparisons. *Journal of the American statistical Association*, 69(347):730–737, 1974.
- [139] H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige. Double window optimisation for constant time visual SLAM. In *2011 International Conference on Computer Vision*, pages 2352–2359, Nov 2011.
- [140] 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*, pages 573–580. IEEE, 2012.

- [141] R. Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [142] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang. Convolutional neural networks for medical image analysis: full training or fine tuning? *IEEE transactions on medical imaging*, 35(5):1299–1312, 2016.
- [143] J.-A. Ting, E. Theodorou, and S. Schaal. Learning an outlier-robust kalman filter. In *European Conference on Machine Learning*, pages 748–756. Springer, 2007.
- [144] L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3d camera tracking. In *Mixed and Augmented Reality, 2004. ISMAR 2004. Third IEEE and ACM International Symposium on*, pages 48–56. IEEE, 2004.
- [145] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proc. of the 25th Int. Conf. on Machine learning*, pages 1096–1103. ACM, 2008.
- [146] G. Vogiatzis and C. Hernández. Video-based, real-time multi-view stereo. *Image and Vision Computing*, (7):434–441.
- [147] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, 2010.
- [148] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [149] B. Williams, G. Klein, and I. Reid. Real-time SLAM relocation. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.
- [150] J. Witt and U. Weltin. Robust Stereo Visual Odometry Using Iterative Closest Multiple Lines. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4164–4171. IEEE/RSJ, 2013.
- [151] G. Zhang, J. H. Lee, J. Lim, and I. H. Suh. Building a 3-D Line-Based Map Using Stereo SLAM. *IEEE Transactions on Robotics*, 31(6):1364–1377, 2015.
- [152] G. Zhang and I. H. Suh. Building a partial 3D line-based map using a monocular SLAM. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1497–1502. IEEE, 2011.

- [153] L. Zhang and R. Koch. Hand-held monocular SLAM based on line segments. In *Machine Vision and Image Processing Conference (IMVIP), 2011 Irish*, pages 7–14. IEEE, 2011.
- [154] L. Zhang and R. Koch. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. *Journal of Visual Communication and Image Representation*, 24(7):794–805, 2013.
- [155] Z. Zhang, C. Forster, and D. Scaramuzza. Active exposure control for robust visual odometry in HDR environments. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3894–3901. IEEE, 2017.
- [156] H. Zhou, B. Ummenhofer, and T. Brox. Deeptam: Deep tracking and mapping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 822–838, 2018.
- [157] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu. StructSLAM : Visual SLAM with Building Structure Lines. *IEEE Transactions on Vehicular Technology*, 9545(c):1–1, 2015.
- [158] K. Zuiderveld. Contrast limited adaptive histogram equalization. In *Graphics gems IV*, pages 474–485. Academic Press Professional, Inc., 1994.