

TESIS DE LA UNIVERSIDAD
DE ZARAGOZA

2023 176

Alejandro Fontán Villacampa

Information-driven navigation

Director/es

Civera Sancho, Javier
Triebel, Rudolph

<http://zaguan.unizar.es/collection/Tesis>

ISSN 2254-7606



Premsas de la Universidad
Universidad Zaragoza



Universidad
Zaragoza

Tesis Doctoral

INFORMATION-DRIVEN NAVIGATION

Autor

Alejandro Fontán Villacampa

Director/es

Civera Sancho, Javier
Triebel, Rudolph

UNIVERSIDAD DE ZARAGOZA
Escuela de Doctorado

Programa de Doctorado en Ingeniería de Sistemas e Informática

2022

Ph.D Thesis



Universidad
Zaragoza



Information - Driven Navigation



Alejandro Fontán Villacampa

Supervised by Javier Civera and Rudolph Triebel

Departamento de Informática e Ingeniería de Sistemas (DIIS)
Escuela de Ingeniería y Arquitectura (EINA)
German Aerospace Center (DLR)

Universidad de Zaragoza

March 29, 2022



Information-Driven Navigation

Alejandro Fontán Villacampa

Supervisors

Javier Civera Sancho	Universidad de Zaragoza, Spain
Rudolph Triebel	DLR, Institute of Robotics and Mechatronics, Germany Technical University of Munich, Germany

Dissertation Committee

Teresa Vidal Calleja	University of Technology Sydney, Australia
Jose María Martínez Montiel	Universidad de Zaragoza, Spain
Margarita Chli	ETZ Zürich, Switzerland

External Examiners

Riccardo Giubilato	DLR, Institute of Robotics and Mechatronics, Germany
Berta Bescós	Facebook Reality Labs, Switzerland

Abstract

In the last years, we have witnessed an impressive progress in the accuracy and robustness of Visual Odometry (VO) and Simultaneous Localization and Mapping (SLAM). This boost in the performance has enabled the first commercial implementations related to augmented reality (AR), virtual reality (VR) and robotics. In this thesis, we developed new probabilistic methods to further improve the accuracy, robustness and efficiency of VO and SLAM. The contributions of our work are issued in three main publications and complemented with the release of SID-SLAM, the software containing all our contributions, and the challenging Minimal Texture dataset.

Our first contribution is an **information-theoretic approach to point selection** for direct and/or feature-based RGB-D VO/SLAM. The aim is to select only the most informative measurements, in order to reduce the optimization problem with a minimal impact in the accuracy. Our experimental results show that our novel criteria allows us to reduce the number of tracked points down to only 24 of them, achieving state-of-the-art accuracy while reducing $10\times$ the computational demand.

Better uncertainty models for visual measurements will impact the accuracy of multi-view structure and motion and will lead to realistic uncertainty estimates of the VO/SLAM states. We derived a novel **model for multi-view residual covariances based on perspective deformation**, which has become a crucial element in our information-driven approach.

Visual odometry and SLAM systems are typically divided in the literature into two categories, feature-based and direct methods, depending on the type of residuals that are minimized. We combined our two previous contributions in the formulation and implementation of **SID-SLAM**, the first full semi-direct RGB-D SLAM system that uses tightly and indistinctly features and direct methods within a complete information-driven pipeline. Moreover, we recorded **Minimal Texture** an RGB-D dataset with conceptually simple but challenging content, with accurate ground truth to facilitate state-of-the-art research on semi-direct SLAM.

Resumen


En los últimos años, hemos presenciado un progreso enorme de la precisión y la robustez de la “Odometría Visual” (VO) y del “Mapeo y la Localización Simultánea” (SLAM). Esta mejora de su funcionamiento ha permitido las primeras implementaciones comerciales relacionadas con la realidad aumentada (AR), la realidad virtual (VR) y la robótica. En esta tesis, desarrollamos nuevos métodos probabilísticos para mejorar la precisión, robustez y eficiencia de estas técnicas. Las contribuciones de nuestro trabajo están publicadas en tres artículos y se complementan con el lanzamiento de “SID-SLAM”, el software que contiene todas nuestras contribuciones, y del “Minimal Texture dataset”.

Nuestra primera contribución es un **algoritmo para la selección de puntos basado en Teoría de la Información** para sistemas RGB-D VO/SLAM basados en métodos directos y/o en características visuales (*features*). El objetivo es seleccionar las medidas más informativas, para reducir el tamaño del problema de optimización con un impacto mínimo en la precisión. Nuestros resultados muestran que nuestro nuevo criterio permite reducir el número de puntos hasta tan sólo 24 de ellos, alcanzando la precisión del estado del arte y reduciendo en hasta 10 veces la demanda computacional.


El desarrollo de mejores modelos de incertidumbre para las medidas visuales mejoraría la precisión de la estructura y movimiento multi-vista y llevaría a estimaciones más realistas de la incertidumbre del estado en VO/SLAM. En esta tesis derivamos un **modelo de covarianza para residuos multi-vista**, que se convierte en un elemento crucial de nuestras contribuciones basadas en Teoría de la Información.

La odometría visual y los sistemas de SLAM se dividen típicamente en la literatura en dos categorías, los basados en *features* y los métodos directos, dependiendo del tipo de residuos que son minimizados. En la última parte de la tesis combinamos nuestras dos contribuciones anteriores en la formulación e implementación de **SID-SLAM**, el primer sistema completo de SLAM semi-directo RGB-D que utiliza de forma integrada e indistinta *features* y métodos directos, en un sistema completo dirigido con información. Adicionalmente, grabamos “**Minimal Texture**”, un *dataset* RGB-D con un contenido visual conceptualmente simple pero arduo, con un *ground truth* preciso para facilitar la investigación del estado del arte en SLAM semi-directo.

Contents

Abstract	iii
<i>Resumen</i>	v
1 Introduction	1
1.1 Towards an Information-Driven Navigation	1
1.2 Related Work	2
1.3 Contributions	10
1.4 Peer-Reviewed Publications	11
2  Information-Driven Direct RGB-D Odometry	12
2.1 Abstract	12
2.2 Introduction	13
2.3 Related work	14
2.4 Notation and fundamentals	15
2.4.1 Photometric model	15
2.4.2 Information metrics	18
2.5 ID-RGBDO- tracking	19

2.5.1	Informative point selection	19
2.5.2	Pose estimation	21
2.6	ID-RGBDO- windowed optimization	22
2.6.1	Keyframe creation	22
2.6.2	Keyframe marginalization	23
2.7	Experimental Results	25
2.7.1	Informative point selection	26
2.7.2	Informative Keyframe Creation	28
2.7.3	Computational Performance	28
2.7.4	Evaluation against SotA baselines	29
2.8	Conclusions and Future Work	30
3	■ A Model for Multi-View Residual Covariances based on Perspective Deformation	31
3.1	Abstract	31
3.2	Introduction	32
3.3	Related Work	34
3.4	Perspective Deformation	35
3.4.1	Preliminaries	35
3.4.2	Surface representation	35
3.4.3	Perspective deformation model	36
3.5	Visual Residual Covariances	38
3.5.1	Implementation Details	39

3.6	Model Validation	41
3.6.1	Geometric covariance	41
3.6.2	Photometric patches	42
3.6.3	Feature-based methods	46
3.7	Experiments	47
3.7.1	Information Metrics	48
3.7.2	Photometric odometry	48
3.7.3	Feature-based SLAM	49
3.8	Conclusions and Future Work	50
4	 SID-SLAM: Semi-Direct Information-Driven RGB-D SLAM	51
4.1	Abstract	51
4.2	Introduction	51
4.3	Related Work	53
4.4	Semi-Direct Model Formulation	54
4.4.1	Informative Point Selection	56
4.4.2	Information-based tracking	58
4.4.3	Bundle Adjustment with Semi-Direct Formulation	58
4.5	SID-SLAM	60
4.5.1	Windowed optimization	60
4.5.2	Loop Closure, Pose Graph Optimization and Global BA	61
4.6	Experiments	62
4.6.1	Results in public RGB-D datasets	62

4.6.2	Results in Minimal Texture Dataset	66
4.7	Conclusions	68
5	DOT: Dynamic Object Tracking for Visual SLAM	69
5.1	Abstract	69
5.2	Introduction	70
5.3	Related Work	71
5.4	DOT	73
5.4.1	Overview	73
5.4.2	Instance Segmentation	74
5.4.3	Camera and Object Tracking	74
5.4.4	Tracking quality, outliers and occlusions	76
5.4.5	Is the object in motion?	77
5.4.6	Mask propagation	79
5.5	Experimental Results	80
5.5.1	Evaluation against baselines	80
5.5.2	Mask propagation	85
5.6	Conclusions	85
6	Conclusion	88
6.1	Summary of Thesis Achievements	88
6.2	Discussion and Future Work	91
6.3	<i>Resumen de los logros de la tesis</i>	92
	Bibliography	94

Chapter 1

Introduction

1.1 Towards an Information-Driven Navigation

Imagine a drone or a rover, equipped with cameras, that have to navigate autonomously on a planetary exploration mission. Or a pair of smart glasses, again with built-in cameras, able to enrich people’s environment with augmented reality (AR) content. Both applications, even being radically different, rely on methods that can estimate in real time the structure of the environment and the motion of the camera from the video stream. We are specifically talking about well-known fields of “Visual Odometry (VO)” and “Simultaneous Localization and Mapping (SLAM)”.

Both vision-based state estimation techniques, VO and SLAM, have experienced a gigantic boost in the number of available methods and open-source codes in recent years, most likely in relation to a substantial improvement of their accuracy and robustness. However, in spite of their respective successes, VO and SLAM are still facing significant challenges. In particular, the application cases mentioned at the beginning of this section, a drone in exploratory tasks and a pair of AR glasses, share one of the main challenges of the state of the art: both demand high standards for accuracy and robustness, while the low-end platforms on which they run impose severe constraints in terms of computational and memory footprints.

Since Shannon established the fundamentals of Information Theory applied to signal processing [Sha48], Information Theory has been extended to a wide variety of disciplines as it allows quantifying and formalizing the analysis of any process related to information. In the field of robotics, it provides metrics to analyze the transmission, processing, extraction and use of information. It has been a relevant topic in the VO/SLAM community, with a considerably large literature, which has produced major advances aiming, among others, (i) to reduce the computational load by finding the most informative/redundant pieces of information, (ii) quantify the goodness of the robot localization or environment representation, (iii) and provide a formal way to make decisions for active navigation. However, to the best of our knowledge, most SLAM baselines apply information metrics partially to independent processes and there is no complete SLAM system that takes full advantage of the potential of Information Theory.

This thesis contributes with novel algorithms and models to enable a full SLAM pipeline that tightly relies on information metrics and accurate uncertainty models. Such SLAM pipeline is thus able to handle all available information in the image in an efficient manner and consequently boosts its accuracy and robustness and reduces its memory footprint. In other words, this thesis is an attempt to push VO and SLAM towards Information-Driven Navigation.

1.2 Related Work

Over the last 15 years, from the earliest real-time demonstrations of SLAM with a monocular camera [DRMS07], the SLAM community has made astonishing progress, witnessing real-world and scientific applications, and enabling the first commercial implementations related to augmented reality (AR), virtual reality (VR) and robotics [APSL08, CCC⁺16, Dav18].

We approach visual SLAM with information-based algorithms [FCT20], which are based on better models for residual covariances [FMCT22], and aim at efficient use of image information with a semi-direct approach (SID-SLAM and Minimal Texture dataset). Table 1.1 summarizes previous work that relates to ours and highlights their main connections to this thesis. The work related to each contribution is detailed in the corresponding chapter.

Scientific Publication	Baseline	Me.	CL	Sensor	Pipel.	Contribution
SID SLAM: Semi-Direct Information-Driven RGB-D SLAM	SID SLAM Dataset	SDi	S	RGB-(D)	SLAM	Tight and independent semi-direct inf.-based point selection. Minimal Texture dataset for semi-direct research.
A model for multi-view residual covariances based on perspective deformation [FMCT22]	Inf. cont.	SDi		RGB-(D)	VO/SLAM	Model feature-based or photometric multi-view residual covariances.
The madmax data set for visual-inertial rover navigation on Mars [MSFV+21]	Dataset			VI/D/S		MADMAX dataset for Mars exploration.
ORB SLAM3: An Accurate Open-Source Library for Visual Inertial, and Multimap SLAM [CER+21]	ORB SLAM3	Ft	S	VI	SLAM	
Dot: dynamic object tracking for visual slam [BFC+21]	DOT	Di	S	RGB-(D)	VO	Instance segment. & multi-view geom. → dynamic object masks. BA solver with single-precision floating-point numbers.
Square Root Bundle Adjustment for Large-Scale Reconstruction [DSCU]		Ft				
Square Root Marginalization for Sliding-Window Bundle Adjustment [DSS+21]		Ft				
Information-driven direct rgb-d odometry [FCT20]	Inf. cont.	Di	S	RGB-(D)	VO	Informative point selection for direct methods.
On the Redundancy Detection in Keyframe-based SLAM [SC19]	Inf. cont.					Remove redundant keyframes with inf.-theoretics & heuristics.
Good feature matching: Toward accurate, robust vo/vslam with low latency [ZV20]	Inf. cont.					
Bad slam: Bundle adjusted direct rgb-d slam [SSP19]	BAD SLAM Dataset	Di	D	RGB-D	SLAM	Intrinsics and depth distortion optimization. ETH3D dataset & RGBD-TUM synt. Well-calibrated benchmark with synchronized global shutter.
Calibration Wizard: A guidance system for camera calibration based on modelling geometric and corner uncertainty [PS19]	Inf. cont.	Ft				Persp. deformation affects multi-view visual residual covariances.
Good feature selection for least squares pose optimization in VO/VSLAM [ZV18]	Inf. cont.	Ft				Approx. NP-hard Max-logDet problem for feature selection.
Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect [YWGC18]	Inf. cont.	SDi				Feature-based: motion bias and pixel discretization. Direct: unmodeled geom. distortions and photom. calibration.
DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes [BFCN18]	DynaSLAM	Ft	S	RGB-D S	SLAM	ORB-SLAM2 + dynamic object detection + background inpainting.
FutureMapping: The computational structure of spatial AI systems [Dav18]	Survey					Spatial AI requirements → Technology real world constraints
Loosely-coupled semi-direct monocular slam [LC18]	LCSD-SLAM	SDi	S	RGB	SLAM	Loose coupling of DSO and ORB-SLAM.
Direct sparse odometry with rolling shutter [SDU+18]		Di	S	RGB	VO	
Online Photometric Calibration of Auto Exposure Video for Realtime Visual Odometry and SLAM [BWC18]	Inf. cont.	Di				
LD SO: Direct sparse odometry with loop closure [GWDC18]	LD SO	SDi	S	RGB	VO	Direct VO + BoW loop closure + pose graph opt.
Stereo DSO: Large-Scale Direct Sparse Visual Odometry with Stereo Cameras [WSC17]	Stereo DSO	Di	S	RGB-S	VO	
Dense Visual SLAM with Probabilistic Surfel Map [YYR17]	PSM SLAM		D	RGB-D	SLAM	Surfel representation + measurement uncertainty modeling.
RGBDTAM: A cost-effective and accurate RGB-D tracking and mapping system [CC17]	RGBDTAM	SDi	SD	RGB-D	SLAM	Direct VO + BoW loop closure + alternating BA.
Monocular visual odometry: Sparse joint optimisation or dense alternation? [PDL17]	Inf. cont.					Sparse joint optimization ~ Dense alternating optimization.
Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras [MAT17a]	ORB SLAM2	Ft	S	RGB-D S	SLAM	Over-insertion & redundant removal of keyframes. Tracking, mapping & relocalization over the same features. Integrates a full photometric calibration. Joint optimization of all model parameters. Photometric residuals weighted with a gradient-dependent term.
Direct sparse odometry [EK17]	DSO	Di	S	RGB	VO	
Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration [DNZ+17]	Bundlefusion	SDi	D	RGB-D	SLAM	
A photometrically calibrated benchmark for monocular visual odometry [EUC16]	Dataset			RGB		Monocular Visual Odometry Dataset
SVO: Semidirect visual odometry for monocular and multicamera systems [FZG+16]	SVO	SDi	S	RGB	VO	
Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age [CCC+16]	Survey					Surveying the surveys, tutorials, open challenges {Long term, Active SLAM, ...} and new {tools, formulations, sensors and learning}.
ElasticFusion: Real-time dense SLAM and light source estimation [WSMG+16]	ElasticFusion	D	D	RGB-D	SLAM	
ElasticFusion: Dense SLAM without a pose graph [WLSM+15]	ElasticFusion	D	D	RGB-D	SLAM	Frame-to-model photom. image alignment for camera tracking. Windowed fused surfel-based model of the environment.
ORB-SLAM: a versatile and accurate monocular SLAM system [MAMT15]	ORB SLAM	Ft	S	RGB	SLAM	Joint optimization of the estimated 3D map and camera trajectory. Residuals weighted as a function of the feature scale.
Real-time large-scale dense RGB-D SLAM with volumetric fusion [WKJ+15]		Di	D	RGB-D	SLAM	
An evaluation of robust cost functions for RGB direct mapping [CC15]		Di				Best performance of error functions that saturate.
Robust reconstruction of indoor scenes [CZK15]	Dataset			Synt.		Augmented ICL-NUIM Dataset
A benchmark for rgb-d visual odometry, 3d reconstruction and slam [HWMD14]	Dataset			Synt.		ICL-NUIM RGB-D Benchmark dataset Sequences with and without sensor noise. Tracking & triangulation: photom. alignment Joint optimization of struct. and motion: feature-based Different geometric dimension of minimization residuals
SVO: Fast semi-direct monocular visual odometry [FPS14]	SVO	SDi	S	RGB	VO	
LSD-SLAM: Large-scale direct monocular SLAM [ESC14]	LSD SLAM	Di	SD	RGB	SLAM	
Semi-dense visual odometry for a monocular camera [ESC13]		Di	SD	RGB	VO	Image regions with significant intensity gradient (semi-dense). Uncertainty-Aware stereo depth map estimation.
Dense visual SLAM for RGB-D cameras [KSC13a]	DVOSLAM	Di	D	RGB-D	SLAM ¹	Photometric approach extended with ICP term. Entropy-based method for Keyframe insertion
Robust odometry estimation for RGB-D cameras [KSC13b]	DVO	Di	D	RGB-(D)	VO	Photometric residuals approximated by a t-distribution . Robustified with the inclusion of a motion prior . Photometric error formulation embedded into a probabilistic framework .
3-D mapping with an RGB-D camera [EHS+13]	3-D SLAM	Ft	S	RGB-(D)	SLAM	
Robust real-time visual odometry for dense RGB-D mapping [WJK+13]		Di	D	RGB-D	VO	ICP extended with photometric term.
Real-time 3D reconstruction at scale using voxel hashing [NZIS13]	Voxel hashing					
Local accuracy and global consistency for efficient visual SLAM [Str12]						SLAM residuals and Jacobians with Lie Algebras.
An evaluation of the RGB-D SLAM system [EHE+12]	RGB-D SLAM	Ft	S	RGB-(D)	SLAM	
A benchmark for the evaluation of rgb-d slam systems [SEE+12a]	Dataset			RGB-D		RGB-D SLAM Dataset and Benchmark Sequences with slow motion, structure, texture and loop closures.
Multi-resolution surfel maps for efficient dense 3D modeling and tracking [SB12]	MRSMap					
Kintinuous: Spatially extended kinectfusion [WKF+12]	Kintinuous	ICP	D	D	Map & Track	Extended KinectFusion to large scenes.
KinectFusion: Real-time dense surface mapping and tracking [NIH+11]	KinectFusion	ICP	D	D	Map & Track	Frame-to-model ICP alignment for camera tracking.
DTAM: Dense tracking and mapping in real-time [NLD11]	DTAM	Di	D	RGB	Map & Track	Frame-to-model photom. image alignment for camera tracking.
Real-time visual odometry from dense RGB-D images [SSC11]		Di	D	RGB-(D)	VO	Frame-to-frame photom. image alignment for camera tracking.
Real-time monocular SLAM: Why filter? [SMD10]		Ft	S			$\frac{\partial(SLAM_{accuracy})}{\partial(\#features)} > \frac{\partial(SLAM_{accuracy})}{\partial(\#frames)}$
Active matching for visual tracking [CD09]		Ft	S	RGB		Sequential Bayesian algorithm for efficient feature search.
The SLAM problem: a survey [APSL08]	Survey					Speed up SLAM for large scale scenarios.
Inverse depth parametrization for monocular SLAM [CDM08]						
Parallel tracking and mapping for small ar workspaces [KM07]	PTAM	Ft	S	RGB	SLAM	Split tracking and mapping into two separate tasks.
MonoSLAM: Real-time single camera SLAM [DRMS07]	MonoSLAM	Ft	S	RGB	SLAM	
Active search for real-time vision [Dav05]		Ft	S	RGB		Theoretical analysis for sequential informative feature search.
Locally Planar Patch Features for Real-Time Structure from Motion [MDR04]		Ft	S	RGB		Points as locally planar 3D patches.
About direct methods [IA99]						

Table 1.1: A brief review of Information-Driven Navigation related work. Labels: Baseline {Information contribution}, Method {Direct, Feature-based, Semi-Direct, ICP}, Computational Load {Sparse, Dense, Semi-Dense}, Sensor {RGB-Monocular, Depth, Stereo, Visual Inertial, RGB-(D) loosely coupled}, Pipeline {Visual Odometry, SLAM, Mapping & Tracking}.

RGB-D VO/SLAM. Due to the origin of photometric methods as dense algorithms, especially in their early days, visual odometry and SLAM techniques based on direct methods have been developed along with algorithms targeting RGB-D sensors.

Steinbrücker et al. [SSC11] implemented a frame-to-frame dense camera tracking based on the minimization of a photoconsistency energy term between stereo pairs of RGB-D images. Kerl et al. [KSC13b] formulated the first probabilistic derivation of a model for dense direct motion estimation. This formulation allowed to include a motion prior from any motion model and suitable sensor, that reduced noise and aided motion estimation convergence with feature-poor images, or in the presence of motion blur or dynamic objects. They also found that the distribution over the photometric residuals is better approximated by a t-distribution than by a Gaussian one.

Newcombe et al. [NLD11] demonstrated that the accuracy of dense alignment can be increased by matching the current image against a scene model, and explored this technique in a subsequent line of work [NIH⁺11, WK⁺12, WJK⁺13, WLSM⁺15, WSMG⁺16].

RGB-D sensors allow the minimization of a geometric error between 3D points, instead of (or in addition to) the minimization of the RGB image error. Newcombe et al. [NIH⁺11] used this type of algorithm known as iterative closest point (ICP) to estimate the camera motion. They observed that in the regime of small displacements image-based errors give better results. However, ICP is more robust to large camera motion and also helps to strengthen photometric approaches in untextured scenes.

Performing an error minimization combining photometric and geometric residuals for camera motion estimation allows to fully and tightly exploit both intensity and depth information from the sensor. Whelan et al. [WJK⁺13] extended their previous ICP-based approach [WJK⁺13] with a photometric term to track the camera's 6DOF motion precisely by frame-rate full image alignment against a entire dense model.

Analogously, Kerl et al. [KSC13a] implemented DVO-SLAM by extending their previous photometric approach [KSC13b] with an ICP term. Full utilization of the dual RGB-D sensor data has been implemented in more recent baselines, such as the work by Concha et al. [CC17] or Schöps et al. [SSP19].

From dense to information-driven sparse SLAM. Thanks to the compact amount of image information retained by keypoints, feature-based SLAM can handle a rigorous joint optimization of the estimated 3D map and camera trajectory [MAMT15]. Dense or semidense reconstructions provide more complete scene reconstructions. However, the large amounts of data make them computationally very expensive and have led them to perform suboptimally alternating between the estimation of motion and structure [SSP19].

Platinsky et al. [PDL17] showed that sparse joint optimization performs similarly to semi-dense or dense alternating optimization. The reason is that the extra amount of data used by dense or semi-dense methods makes up for the loss in accuracy coming from the efficient but suboptimal alternating optimization. Therefore, the scientific challenge became how to maximize the amount of information processed while minimizing the memory and computational footprint.

Reducing the computational demand of dense techniques can be handled with more efficient approaches to algorithms. Demmel et al. [DSS⁺21, DSCU] proposed a formulation for solving large-scale bundle-adjustment problems with single-precision floating-point numbers. They achieved an accuracy equivalent to that of the commonly used Schur complement trick, but could handle larger amounts of memory in dense problems.

Removing redundant information from the bundle adjustment optimization is another approach to reduce the computational demand of dense algorithms. Strasdat et al. [SMD10] showed how, in order to increase the accuracy of monocular SLAM, it is more profitable to increase the number of features than the number of frames. Engel et al. [ESC13, ESC14] reduced the amount of data of dense photometric visual odometry by using only those pixels that lay in image regions with significant intensity gradient (semi-dense). Moreover, they showed in [EKC17] how the joint optimization of all model parameters can be performed in a direct sparse odometry pipeline running in real time on a CPU.

Certainly one of the most elegant and advantageous lines of work to reduce the computational footprint of SLAM is their combination with Information Theory metrics. Feature descriptors have seen a extense development since their first implementations. The high degree of consistency reached by feature matching techniques has allowed an extensive research in information-based point selection. Davison [Dav05] introduced a simulated theoretical analysis for sequential feature search guided by expected Shannon information gain. Based on this analysis, Chli et al. [CD09] contributed Active Matching, a sequential Bayesian algorithm for efficient feature

search, that was able to run with real data dealing with discrete multiple hypotheses which arise due to matching ambiguity. Zhao et al. [ZV18] [ZV20] found that maximizing the logarithm of the determinant of the Information Matrix (Max-logDet) in a pose optimization problem performed the best to guide the feature selection. To that end, they introduced an efficient algorithm for approximately solving the NP-hard Max-logDet problem that significantly improved the accuracy of pose tracking, while introducing little overhead.

We address dense SLAM sparsification with information-driven algorithms [FCT20] based on better models for residual covariances [FMCT22], that look for the most informative image landmarks (either photometric or feature-based) and keyframes in the scene.

Well-founded covariance estimates. Most feature-based VO/SLAM systems consider photometric patches/features as planar surfaces in the image space and set a constant value for their visual covariance. Molton et al. [MDR04] considered points as locally planar 3D patches in a Structure-from-Motion setup. Peng et al. [PS19] proposed an approach for intrinsic camera calibration where they took into account, in a heuristic manner, the influence that transformations apply to image patches when they are viewed from another viewpoint. Engel et al. [EKC17] considered gradient weighting of photometric residuals, and Mur et al. [MAMT15] weighted reprojection residuals as a function of feature scale.

Yang et al. [YWGC18] presented an evaluation method for challenges in monocular visual odometry. They evaluated to what extent photometric calibration, motion bias and rolling shutter influenced feature-based and photometric approaches. They concluded that feature-based methods are more sensitive to pixel discretization artifacts and they suffer a larger performance bias when running forwards and backwards. For direct methods, Yang et al. found that are more affected by unmodeled geometric distortions and by the lack of a photometric calibration. These challenges have been addressed in different works: Engel et al. [EUC16] contributed a photometrically calibrated dataset, Bergmann et al. [BWC18] proposed an online photometric calibration for VO and SLAM, and Schubert et al. [SDU⁺18] extended DSO [EKC17] to work under rolling shutter effects.

Our paper [FMCT22] introduced a general model for the covariance of the visual residuals formulated as a combination of geometric and photometric noise sources. Our key novel contribution is the derivation of a term modelling how local 2D patches suffer from perspective deformation when imaging 3D surfaces around a point.

If one takes a step forward, in order to reduce SLAM computational footprint to its minimum, uncertainties that arise from noise sources such “dynamic objects” or “illumination changes” need to be tackled. We required novel algorithms and tools that estimate not only uncertainties from geometric sources (such as an RGB-D sensor) but also those associated to more complex scene behaviours. Based on the work of Bescós et al. [BFCN18] we developed Dynamic Object Tracking (DOT) [BFC⁺21], a front-end that combines instance segmentation and multi-view geometry to generate masks for dynamic objects in order to allow SLAM systems based on rigid scene models to avoid such image areas in their optimizations.

Semi-direct methods. Combining features and direct methods is widespread in the literature, however it is commonly performed on only specific and isolated parts of the systems in a loosely coupled manner. The goal is to exploit the complementarity of photometric and feature-based methods (see the discussion in Table 1.2 and section 4.2), and the challenge is to achieve this without compromising efficiency, accuracy or robustness.

Specifically, some baselines perform the combination by assigning the most appropriate task to each method. Forster et al. [FPS14, FZG⁺16] used photometric alignment for tracking and pixel triangulation, and feature-based joint optimization of structure and motion. In their work they exploited the difference in the geometric dimension of both minimization residuals. Similarly, Lee et al. [LC18] combines photometric bundle adjustment of the local structure and motion [EKC17] and geometric bundle adjustment for larger optimization windows [MAMT15].

The work from Gao et al. [GWDC18] adds to a direct VO thread a bag-of-words loop closure and the optimization of a co-visibility graph of keyframe poses. Schöps et al. [SSP19] and Concha et al. [CC17] use a similar approach, but in these cases the map optimization is done by an alternating direct Bundle Adjustment.

RGB-D Datasets & Benchmarks. The evaluation and comparison between scientific approaches in SLAM are commonly performed in public datasets and benchmarks.

Sturm et al. [SEE⁺12a] contributed the *RGB-D SLAM Dataset and Benchmark*. The RGB-D data was recorded with accurate ground truth camera poses in an office environment and an industrial hall. It was motivated to serve as a benchmark for the evaluation of RGB-D SLAM systems, providing: slow motion for debugging, scenes with varying degrees of structure and texture, and trajectories with and without loop closures.

	Features	Direct methods
Basic data	<ul style="list-style-type: none"> • Salient keypoints with invariant descriptors. 	<ul style="list-style-type: none"> • Image’s pixel-level intensities.
Image information	<ul style="list-style-type: none"> • Exploits a small subset of the information. ✗ • Robust to relatively large illumination and viewpoint changes. ✓ • Suffer from motion bias and pixel discretization. ✗ • Need of pre-processing steps. ✗ 	<ul style="list-style-type: none"> • Make use of all intensity gradients. ✓ • More sensitive to unmodeled geometric distortions. ✗ • Need photometric calibration ✗ • No need of pre-processing steps. ✓
Data association	<ul style="list-style-type: none"> • Performed independently for each feature at frame rate. ✗ • Detectors are optimized for speed rather than precision. ✗ • Need robust estimation techniques. ✗ 	<ul style="list-style-type: none"> • Direct does not need a prior step of data association since data is implicitly associated in the geometry model. ✓ • Track weak corners/edges in little/high-frequency textures. ✓
Minimization error	<ul style="list-style-type: none"> • 2D Reprojection error. • Corner alignment happens in the two image directions. • A large convergence baseline that allows for stronger movements. ✓ 	<ul style="list-style-type: none"> • 1D Photometric Error • The alignment of an edge is restricted to the normal direction of the edge. • Small convergence baseline. ✗
BA	<ul style="list-style-type: none"> • Full joint optimization with BA algorithms are widely used with sparse features. ✓ 	<ul style="list-style-type: none"> • The extra amount of data used by dense methods makes up for the loss in accuracy coming from the efficient but suboptimal alternating optimization. ✓
Computational Cost	<ul style="list-style-type: none"> • Low speed due to feature extraction and matching at every frame. ✗ • Reduction to sparse keypoints speeds up computation time enormously in joint optimization of the estimated 3D map and camera trajectory (BA). ✓ 	<ul style="list-style-type: none"> • Dense or semidense reconstruction of the environment are computationally expensive. ✗ • Approximations such as pose graph optimization or deformable geometry. ✗
Optimality	<ul style="list-style-type: none"> • Small memory footprint. Just save features and descriptors. ✓ 	<ul style="list-style-type: none"> • Needs big memory allocation. Saves all full images. ✗
Consistency	<ul style="list-style-type: none"> • High degree of consistency thanks to low matching uncertainties, allowing them to be more easily implemented in real-world applications and combined with other sensors. ✓ 	<ul style="list-style-type: none"> • Lack of well-founded covariance estimates from photometric VO. ✗ • Difficult fusion with complementary sensors. ✗
Drift	<ul style="list-style-type: none"> • Relocalization capabilities. ✓ • Long feature tracks with minimal feature drift. ✓ • Loop closure. ✓ 	

Table 1.2: **Features vs direct methods.** Check section 4.2 for the full discussion.





ICL-NUIM by Handa et al. [HWMD14] is a collection of synthetically generated handheld RGB-D camera sequences, with and without introducing sensor noise in both RGB and depth data. The two indoor environment models (living room and office) were extended by Choi et al. [CZK15] among other extensions, with a more realistic noise model that resulted in noisier depth images.

Schops et al. [SSP19] contributed the *ETH3D* dataset and benchmark. A challenging set of sequences with highly accurate calibrated hardware (e.g. synchronized global shutter RGB and depth cameras) and accurate ground truth demonstrated the excellent performance of their dense photometric SLAM approach. In addition, they created a synthetic version of 7 sequences from the TUM RGB-D dataset by performing dense 3D reconstructions and rendering them with their ground truth. These sequences are available under four variations by adding rolling shutter and asynchronous frames, both individually and combined.

Meyer et al. [MSFV⁺21] recorded *The madmax data set for research on visual-inertial rover navigation on Mars*, with a complete sensor unit that provides time-stamped recordings from monochrome stereo cameras, a color camera, omnidirectional cameras in stereo configuration, and from an inertial measurement unit. Evaluation of the state-of-the-art ORB-SLAM2 [MAT17a] and VINS-MONO [QLS18] systems has shown that there is room for improvement in visual SLAM in low-texture planetary exploration, so this dataset represents a unique tool for future research into visual techniques such as semi-direct SLAM.




Precisely, we recorded *Minimal Texture dataset* to facilitate state-of-the-art research on semi-direct SLAM, particularly: (i) a better understanding of visual uncertainties of both features and photometric approaches, (ii) the efficient use of all the information on the image which maximizes SLAM robustness and reduces its computational footprint.

1.3 Contributions

-  **An information-theoretic approach to point selection** for direct RGB-D odometry [FCT20]: The aim is to select only the most informative measurements, in order to reduce the optimization problem with a minimal impact in the accuracy. Our results show that our novel information-based selection criteria allows us to reduce the number of points down to only 24 of them, achieving an accuracy similar to the state of the art while reducing 10× the computational demand.
-  **A model for Multi-View Residual Covariances based on Perspective Deformation** [FMCT22]: The core of our approach is the formulation of the residual covariances as a combination of geometric and photometric noise sources. And our key novel contribution is the derivation of a term modelling how local 2D patches suffer from perspective deformation when imaging 3D surfaces around a point. These add up to an efficient and general formulation which improves the accuracy of both feature-based and direct methods, and can also be used to estimate more accurate measures of the state entropy.
-  **SID-SLAM**. The release of a full SLAM framework for RGB-D cameras. Our main contribution is a semi-direct approach that, for the first time, combines tightly and indistinctly photometric and feature-based image measurements. Our evaluation on several public datasets shows that we further improve state-of-the-art performance regarding accuracy, robustness and computational footprint in CPU real time.
-  **Minimal Texture**. We recorded this new dataset to facilitate state-of-the-art research on semi-direct SLAM, particularly: (i) a better understanding of visual uncertainties of both features and photometric approaches, (ii) the efficient use of all the information on the image which maximizes SLAM robustness and reduces its computational footprint.
- **Dynamic Object Tracking** [BFC⁺21]: a front-end that combines instance segmentation and multi-view geometry to generate masks for dynamic objects in order to allow SLAM systems based on rigid scene models to avoid such image areas in their optimizations.

1.4 Peer-Reviewed Publications

The core of the research developed in this thesis relies on the following peer-reviewed publications:

-  Alejandro Fontan, Javier Civera, and Rudolph Triebel. **Information-driven direct rgb-d odometry.** In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4929–4937, 2020. (**CVPR oral presentation**)
[Paper](#) [Oral Presentation](#) [Video](#)
-  Alejandro Fontan, Laura Oliva, Javier Civera, and Rudolph Triebel. **A model for multi-view residual covariances based on perspective deformation.** In 2022 IEEE Robotics and Automation Letters (RA-L) and International Conference on Robotics and Automation (ICRA).
[Paper](#) [Video](#)
-  Alejandro Fontan, Riccardo Giubilato, Laura Oliva, Javier Civera, and Rudolph Triebel. **SID-SLAM: Semi-Direct Information-Driven RGB-D SLAM.** *This work is under review in European Conference on Computer Vision (ECCV), 2022.*
- Irene Ballester, Alejandro Fontan, Javier Civera, Klaus H Strobl, and Rudolph Triebel. **Dot: dynamic object tracking for visual slam.** In 2021 IEEE International Conference on Robotics and Automation (ICRA), pages 11705–11711. IEEE, 2021.
[Paper](#) [Video.](#)
- Lukas Meyer, Michal Smisek, Alejandro Fontan Villacampa, Laura Oliva Maza, Daniel Medina, Martin J Schuster, Florian Steidle, Mallikarjuna Vayugundla, Marcus G Muller, Bernhard Rebele, et al. **The madmax data set for visual-inertial rover navigation on mars.** Journal of Field Robotics (JFR), 2021.
[Paper](#) [Webpage](#)

Chapter 2

■ Information-Driven Direct RGB-D

Odometry

2.1 Abstract

This chapter presents an information-theoretic approach to point selection for direct RGB-D odometry. The aim is to select only the most informative measurements, in order to reduce the optimization problem with a minimal impact in the accuracy. It is usual practice in visual odometry/SLAM to track several hundreds of points, achieving real-time performance in high-end desktop PCs. Reducing their computational footprint will facilitate the implementation of odometry and SLAM in low-end platforms such as small robots and AR/VR glasses. Our experimental results show that our novel information-based selection criteria allows us to reduce the number of tracked points an order of magnitude (down to only 24 of them), achieving an accuracy similar to the state of the art (sometimes outperforming it) while reducing $10\times$ the computational demand.

2.2 Introduction

In the last years, we have witnessed an impressive progress in the accuracy and robustness of visual odometry and Simultaneous Localization and Mapping (SLAM) [MAMT15, PFC⁺15, MAT17a, EKC17, QLS18]. This boost in the performance has enabled the transfer of visual odometry and SLAM to several commercial products related to augmented reality (AR), virtual reality (VR) and robotics.

In spite of their respective successes, visual odometry and SLAM are still facing significant challenges. The high computational demand of the state of the art is among the most critical ones for a widespread use in real applications. The embodiment of localization and mapping algorithms into small robotic/AR/VR platforms will impose constraints on their computational and memory footprints [Dav18]. Most algorithms currently require a hardware that exceeds the capabilities of many existing and foreseeable platforms.

In this work we aim to drastically reduce the computational load of direct RGB-D odometry with a negligible loss in accuracy. For that, we propose a novel and efficient information-based criterion to keep only the most informative point in the local Bundle Adjustment and pose tracking optimizations. We implemented a RGB-D odometry (that we denote ID-RGBDO) and evaluated our approach in the TUM dataset, demonstrating that we can achieve substantial reductions in the number of tracked features without noticeably degrading the accuracy. We outperform the naive selection approaches used in the literature, that mainly select points on a grid to maximize coverage.

Observe the two estimated trajectories in Figure 2.1, one tracking the 24 most informative points and the second one 500 points –a reasonable number in the state of the art. Notice that they have almost the same accuracy, but the one using 24 points requires roughly $10\times$ less computation. Our proposed information criteria are able to select the small set of highly informative points that makes this possible.

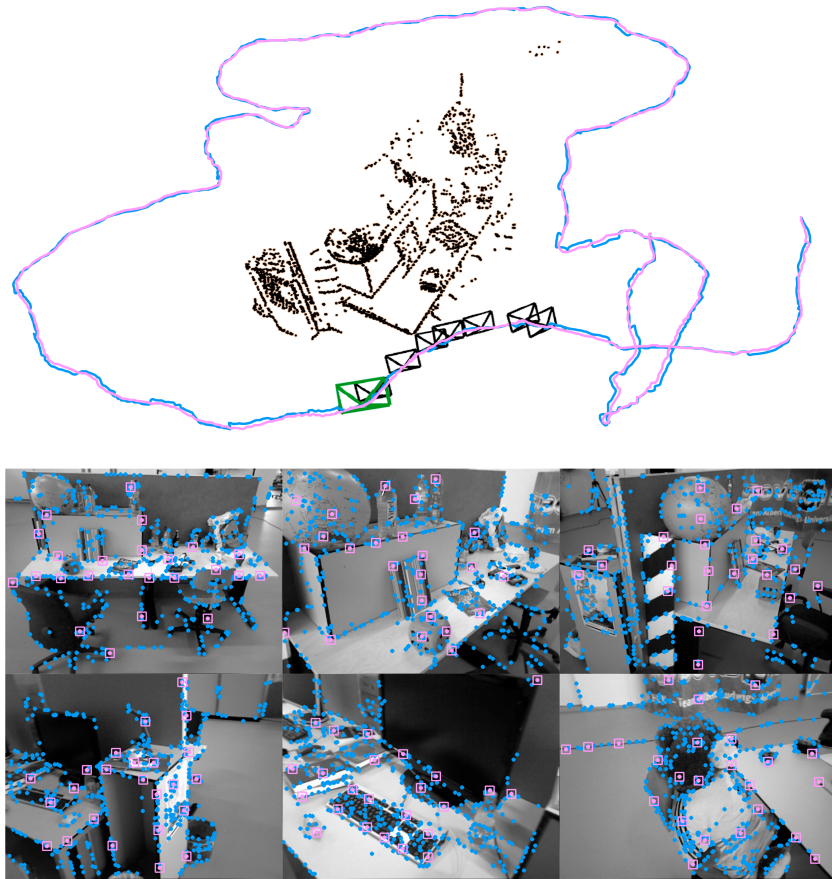


Figure 2.1: **Top:** Trajectories and maps estimated by our RGB-D odometry (ID-RGBDO) in two cases: tracking 500 image points (blue), and tracking only the 24 most informative points (magenta) with considerable computational savings. The difference between the two is almost unnoticeable. **Bottom:** Sample frames and tracks for the 500 points case (blue dots) and the 24 most informative ones (magenta squares).

2.3 Related work

Graph reduction is a relevant topic in the SLAM community, with a considerably large literature [HKL13, CKB⁺14, HHW⁺18]. We focus here on the main approaches using information theory and in particular those developed for visual SLAM.

Information was first used in EKF-based monocular SLAM in [Dav05] in order to guide sequential search. Based on it, [CD08, CD09, HCSD10] introduced a multi-hypothesis formulation able to address ambiguous cases robustly. An information analysis of filtering and Bundle Adjustment was used in [SMD12] to prove the advantages of the latter. Up to our knowledge, ours is the first work that addresses information in a direct odometry framework.

Information-based approaches have been also used in laser-based SLAM. [IPAC09] proposes

a method to add only non-redundant and informative links to a pose graph, and [KS12] uses mutual information to remove low informative laser scans from the graph. The approach in [CICD15] is able to reduce not only poses, but also landmarks, based on information theoretic criteria. [VDWB11, WXLH13] use the Kullback-Leibler divergence to sparsify a SLAM graph.

2.4 Notation and fundamentals

Our direct information-driven odometry minimizes the photometric reprojection error in a sliding window of frames. Our formulation, based on direct Bundle Adjustment and Tracking, is related to recent approaches to direct visual odometry and SLAM, namely [EKC17, CC17, KSC13b, KSC13a]. However, we implemented ID-RGBDO in order to have a higher degree of control in the evaluation. Notice, in any case, that our contribution can be applied to any RGB-D odometry system and should give similar improvements.

This section will cover the necessary background and notation, and the specifics of our RGB-D odometry and contributions will be detailed in Section 2.5 (camera pose tracking) and Section 2.6 (sliding-window Bundle Adjustment).

2.4.1 Photometric model

Point representation. For a point p , its image coordinates are denoted as $\mathbf{p} = \begin{bmatrix} p_u & p_v \end{bmatrix}^\top \in \mathbb{R}^2$ and its inverse depth in the camera frame as $d \in \mathbb{R}$. For its photometric appearance, we use a set of intensity values spread in a patch centered in \mathbf{p} [EKC17].

Keyframe representation. A keyframe j is defined by its RGB-D channels, its 6DOF camera pose as a transformation matrix $\mathbf{T} \in \mathbf{SE}(3)$, two brightness parameters $\{a_j, b_j\}$ and a set of reference points to track. The Lie-algebras pose-increments $\widehat{\mathbf{x}}_{\mathfrak{se}(3)} \in \mathfrak{se}(3)$, with $\widehat{\cdot}_{\mathfrak{se}(3)}$ being the mapping operator from the vector to the matrix representation of the tangent space [Str12], are expressed as a vector $\mathbf{x} \in \mathbb{R}^6$. During the optimization, we update the transformations at step (k) using left matrix multiplication and the exponential map operator $\exp(\cdot)$, i.e.,

$$\mathbf{T}^{(k+1)} = \exp(\widehat{\mathbf{x}}_{\mathfrak{se}(3)}) \cdot \mathbf{T}^{(k)}. \quad (2.1)$$

Residual function. The photometric residual r_i of an image point p_i in a frame i is the intensity difference with the corresponding point in a reference keyframe j , combined with an affine brightness transformation and a robust norm [EKC17]

$$r_i = \left\| e^{-a_j}(I_j(\mathbf{p}_j) - b_j) - e^{-a_i}(I_i(\mathbf{p}_i) - b_i) \right\|_{\gamma}. \quad (2.2)$$

Although some works use the t-distribution [KSC13a, KSC13b], we observed a higher accuracy using the Huber norm (as in [EKC17]) and saturating large values (as in [CC15]).

The image points \mathbf{p}_i and \mathbf{p}_j are related by

$$\mathbf{p}_i = \Pi(\mathbf{R}\Pi^{-1}(\mathbf{p}_j, d_j) + \mathbf{t}), \quad (2.3)$$

where $\Pi(\mathbf{P})$ projects in the image plane the point \mathbf{P} in the camera frame; and $\Pi^{-1}(\mathbf{p}, d)$ back-projects the image point with coordinates \mathbf{p} at inverse depth d . $\mathbf{R} \in \mathbf{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$ are the relative rotation and translation between keyframe j and frame i .

Optimization. We do Gauss-Newton optimization, that can be written as

$$(\mathbf{J}^T \boldsymbol{\Sigma}_r^{-1} \mathbf{J}) \mathbf{y} = -\mathbf{J}^T \boldsymbol{\Sigma}_r^{-1} \mathbf{r}, \quad (2.4)$$

where the rows of the matrix $\mathbf{J} = \begin{bmatrix} \mathbf{J}_x & \mathbf{J}_d & \mathbf{J}_{a,b} \end{bmatrix} \in \mathbb{R}^{n \times m}$ contains the derivatives of the residual function (equation (2.2)) with respect to the Lie-algebra increments \mathbf{J}_x , the point inverse depths \mathbf{J}_d and the photometric parameters $\mathbf{J}_{a,b}$. The diagonal matrix $\boldsymbol{\Sigma}_r \in \mathbb{R}^{n \times n}$ contains the covariances σ_r^2 of the photometric residuals. The residual vector $\mathbf{r} \in \mathbb{R}^n$ stacks the n individual residuals to minimize. $\mathbf{y} \in \mathbb{R}^m$ stands for the state correction containing the increments for poses, inverse depths and photometric parameters.

Residual covariance. Our residual covariance σ_r^2 includes the impact from geometry and appearance. We propose to model it by multiplying a photometric term σ_{Φ}^2 with a geometric one $h(\delta A)$ that comes from projecting a differential area surrounding the 3D point:

$$\sigma_r^2 = h(\delta A) \cdot \sigma_{\Phi}^2. \quad (2.5)$$

Figure 2.2 illustrates how the differential area around a point changes with the viewpoint. This change δA can be modeled as the determinant of the derivative of the image point \mathbf{p}_i in frame i with respect to the coordinates \mathbf{p}_j of the corresponding point in a reference keyframe j :

$$\delta A = \left| \frac{\partial \mathbf{p}_i}{\partial \mathbf{p}_j} \right|. \quad (2.6)$$

With this, we define the geometric weight $h(\delta A)$ as the following function, that penalizes the residual covariance for large perspective distortions

$$h(\delta A) = e^{c_h(\delta A - 1)^2}, \quad (2.7)$$

where c_h is a constant to ponder the influence of the model.

The photometric term σ_{Φ}^2 is computed from a first order propagation of the inverse depth covariance σ_d^2

$$\sigma_{\Phi}^2 \approx \left[\left(g_u \frac{\partial p_u}{\partial d} \right)^2 + \left(g_v \frac{\partial p_v}{\partial d} \right)^2 \right] \sigma_d^2, \quad (2.8)$$

where the intensity gradients $\begin{bmatrix} g_u & g_v \end{bmatrix}$ come from a first-order Taylor expansion of the intensity in the vicinity of \mathbf{p}

$$I(\mathbf{p} + \delta \mathbf{p}) \approx I(\mathbf{p}) + \begin{bmatrix} g_u & g_v \end{bmatrix} \begin{bmatrix} \delta p_u \\ \delta p_v \end{bmatrix}. \quad (2.9)$$

Using the stereo model for RGB-D cameras based on structured light patterns, and assuming a focal length f and a baseline b , the inverse depth error covariance σ_d is [CC17]

$$\sigma_d = \frac{1}{fb} \sigma_{px}, \quad (2.10)$$

where σ_{px} is the disparity error.

2.4.2 Information metrics

Information theory provides a mean to quantify and formalize all processes related with information. In the context of SLAM the special case of multivariate Gaussians is comprehensively well founded [Dav05, CD08]. The information-driven formulation proposed in this chapter is based on the following classical information metrics.

Differential entropy of a k -dimensional Gaussian distribution $\mathbf{X} \sim \mathcal{N}_k(\boldsymbol{\mu}_X, \boldsymbol{\Sigma}_X)$. It can be seen as the expected information content of a future event, given the set of possible results and their probability distribution [CD08]

$$H(\mathbf{X}) = \frac{1}{2} \log((2\pi e)^k |\boldsymbol{\Sigma}_X|). \quad (2.11)$$

Entropy reduction, which is the relative difference between two Gaussian distributions

$$\Delta H(\mathbf{X}, \mathbf{Y}) = H(\mathbf{X}) - H(\mathbf{Y}) = \frac{1}{2} \log \frac{|\boldsymbol{\Sigma}_X|}{|\boldsymbol{\Sigma}_Y|}, \quad (2.12)$$

that is, how much more accuracy is obtained by measuring \mathbf{Y} instead of \mathbf{X} [SMD12].

Conditional covariance. Assuming $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^M$ are combined in a joint Gaussian $\mathbf{Z} \sim \mathcal{N}(\boldsymbol{\mu}_Z, \boldsymbol{\Sigma}_Z)$,

$$\boldsymbol{\Sigma}_Z = \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{bmatrix}, \quad (2.13)$$

the conditional covariance $\boldsymbol{\Sigma}_{x|y}$ of \mathbf{x} given \mathbf{y} , is the Schur complement of $\boldsymbol{\Sigma}_{yy}$ in $\boldsymbol{\Sigma}_Z$:

$$\boldsymbol{\Sigma}_{x|y} = \boldsymbol{\Sigma}_x^* = \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1} \boldsymbol{\Sigma}_{yx}. \quad (2.14)$$

Mutual information between two random variables. It measures how much knowing one of the variables reduces the uncertainty about the other [SC19]:

$$MI(\mathbf{x}, \mathbf{y}) = \frac{1}{2} \log \frac{|\boldsymbol{\Sigma}_{xx}|}{|\boldsymbol{\Sigma}_x^*|}. \quad (2.15)$$

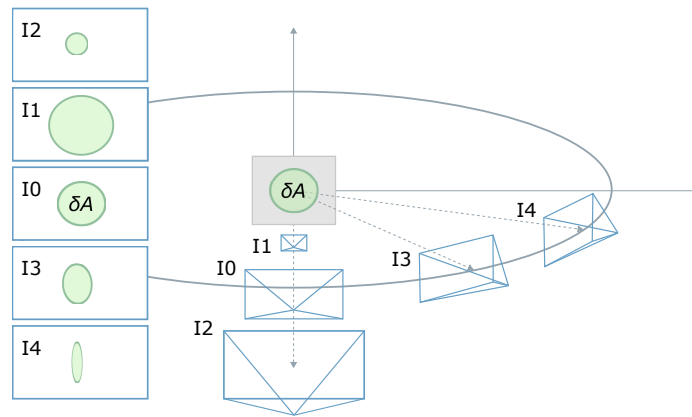


Figure 2.2: Illustration of the projective distortion of a differential 2D patch δA .

Throughout the chapter the entropy is measured in absolute numbers of bits (*i.e.*, \log stands for base-2 logarithm).

2.5 ID-RGBDO- tracking

We now apply to direct RGB-D pose tracking the ideas above, presented in this section theoretically and evaluated experimentally in Section 2.7.1.

2.5.1 Informative point selection

Most direct methods are either dense or semi-dense approaches, aiming to use as many pixels as possible. In order to achieve real-time performance, they rely on a high-end computational platform or make use of sub-optimal approximations.

Sparse direct methods, on the contrary, reduce the number of points by extracting those with a significant photometric gradient (Step 1) and widely spread across the image (Step 2). These heuristics work reasonably well in a wide array of scenarios, although several aspects are left unexplored: Are we reaching the lowest possible error given our data? Are we using redundant information and hence wasting computation? Is there enough visual information for the problem to be well conditioned at all times? Our proposal is to add an algorithm (Step 3) that selects points in a manner that, together with the previous two conditions, maximizes the entropy of the camera pose.

The camera pose entropy depends on the determinant of its covariance matrix Σ_x , as shown in equation (2.11). Each point p contributes with $\Delta_p \Lambda_x$ to the information matrix Λ_x , that can be obtained as the sum of the Jacobian autoprodut for the whole set of points \mathcal{P}

$$\Lambda_x = \Sigma_x^{-1} = \sum_{p \in \mathcal{P}} \Delta_p \Lambda_x = \sum_{p \in \mathcal{P}} \mathbf{j}_{x,p}^T \sigma_r^{-2} \mathbf{j}_{x,p}, \quad (2.16)$$

where $\mathbf{j}_{x,p}$ is the row of the Jacobian \mathbf{J}_x that corresponds to the photometric residual of point p .

The addition of point p also yields to a variation of the information matrix determinant $\Delta_p |\Lambda_x|$, that has the very satisfying property¹ that can be expressed individually per point, depending on the p^{th} row of the Jacobian $\mathbf{j}_{x,p}$ and the current adjoint information matrix Λ_x^{adj} :

$$\begin{aligned} \Delta_p |\Lambda_x| &= |\Lambda_x + \mathbf{j}_{x,p}^T \sigma_r^{-2} \mathbf{j}_{x,p}| - |\Lambda_x| \\ &= |\Lambda_x| |\mathbf{I} + \Lambda_x^{-1} \mathbf{j}_{x,p}^T \sigma_r^{-2} \mathbf{j}_{x,p}| - |\Lambda_x| \\ &= |\Lambda_x| (1 + \sigma_r^{-2} \mathbf{j}_{x,p} \Lambda_x^{-1} \mathbf{j}_{x,p}^T) - |\Lambda_x| \\ &= \sigma_r^{-2} \mathbf{j}_{x,p} \Lambda_x^{adj} \mathbf{j}_{x,p}^T. \end{aligned} \quad (2.17)$$

Based on this, our algorithm works as follows. We start from a pre-filtered set of high-gradient pixels by using a grid with a region-adaptative gradient threshold (as in [EKC17]). We prioritize points that belong to Canny edges (as in [CC17]) but also keep some points in areas with weaker gradient (Step 2). From here we follow Algorithm 1. We choose for each degree of freedom (each of the six columns of \mathbf{J}_x) the image point p with maximum derivative, and build with them an initial information matrix. We then iteratively select the point that maximizes the following function (Step 3)

$$f(p \in \mathcal{P}, z, \Lambda_x) = \Delta_p |\Lambda_x| + \frac{1}{c_z (z_p - z)^2 + 1}. \quad (2.18)$$

The first addend in the function takes into account the increment of information described above. The second one contributes to spread the points in the image, in order to compensate

¹For simplicity we applied a consequence of the Sylvester's determinant theorem $|(I_m + cr)| = 1 + rc$.

Algorithm 1 Informative point selection.

```

1: function SELECT INF. POINTS ( $m, \mathcal{P}, \mathbf{J}_x$ )
2:                                     ▷  $m$  = number of points to be selected
3:                                     ▷  $\mathcal{P}$  = set of available points
4:    $\mathcal{Q} \leftarrow \emptyset$                                      ▷  $\mathcal{Q}$  = set of selected points
5:    $\mathbf{\Lambda}_x \leftarrow \mathbf{0}$                                ▷ Init. Information matrix
6:   for  $k \leftarrow 1$  to DOF do                               ▷ DOF = 6
7:      $i \leftarrow \arg \max (\mathbf{j}_{x,p}[k])$ 
8:      $\mathbf{\Lambda}_x \leftarrow \mathbf{\Lambda}_x + \Delta_p \mathbf{\Lambda}_x(\mathcal{P}[i])$ 
9:      $\mathcal{Q} \leftarrow \mathcal{Q} \cup \mathcal{P}[i]$                                ▷ Add selected point
10:     $\mathcal{P} \leftarrow \mathcal{P} - \mathcal{P}[i]$ 
11:  end for
12:                                     ▷ Informative selection
13:   $z \leftarrow$  image border
14:  while ( $\mathcal{P} \neq \emptyset$  &  $\dim(\mathcal{Q}) < m$ ) do
15:     $i \leftarrow \arg \max (f(\mathcal{P}, z, \mathbf{\Lambda}_x))$                                ▷ Most inf. point
16:     $\mathbf{\Lambda}_x \leftarrow \mathbf{\Lambda}_x + \Delta_p \mathbf{\Lambda}_x(\mathcal{P}[i])$ 
17:     $\mathcal{Q} \leftarrow \mathcal{Q} \cup \mathcal{P}[i]$ 
18:     $\mathcal{P} \leftarrow \mathcal{P} - \mathcal{P}[i]$ 
19:     $z \leftarrow z - \Delta z$ 
20:  end while
21:  return  $\mathcal{Q}$ 
22: end function

```

for effects that are not modeled in the projection function. This last expression increases its value when the radial coordinate z_p of a point p approaches z . z is initialized at the image border and its value is reduced by Δz for each selected point until reaching the principal point. c_z models the importance of this second term with respect to the information increment of each point.

2.5.2 Pose estimation

With our selected set of points, we aim to find the motion $\Delta \mathbf{x}$ between the closest keyframe and the current frame, that minimises the photometric residual vector \mathbf{r} (see equation (2.2)). This optimization is initialized with a constant velocity model and a multi-scale pyramid image to aid convergence.

The addition of a kinematic model has been extensively used in odometry and SLAM. [KSC13b] showed that adding a motion prior in direct odometry helps in cases such as lack of texture, motion blur or dynamic content. The motion estimation with such a prior can be written as

$$(\mathbf{J}_x^T \boldsymbol{\Sigma}_r^{-1} \mathbf{J}_x + \boldsymbol{\Sigma}_m^{-1}) \Delta \mathbf{x} = -\mathbf{J}_x^T \boldsymbol{\Sigma}_r^{-1} \mathbf{r} + \boldsymbol{\Sigma}_m^{-1} (\mathbf{x}_{t-1} - \mathbf{x}_t^{(k)}), \quad (2.19)$$

where \mathbf{x}_{t-1} and $\mathbf{x}_t^{(k)}$ are the camera speeds for the previous frame and the last iteration of the current frame respectively. The diagonal covariance matrix $\boldsymbol{\Sigma}_m \in \mathbb{R}^{6 \times 6}$ models the strenght of the motion prior. As explained in [KSC13b], assigning high values to this covariance matrix decreases the influence of the motion prior with respect to the image residuals and vice versa. Tuning the values of the matrix is left to the knowledge of the agent motion or the availability of another type of sensor (such as an IMU).

As in [EKC17], we consider outliers and discard those points whose photometric error exceeds three times the standard deviation of the distribution. This reduces the effect that occlusions and false matches have on the accuracy and robustness of the odometry.

2.6 ID-RGBDO- windowed optimization

2.6.1 Keyframe creation

There are several different strategies to select keyframes from an image sequence, with the aim of estimating a local map. Conservative strategies privilege the use of already existing keyframes before constituting a new one. Only if there is no previous candidate with enough overlap, the system assumes that a new area is being explored and creates a new keyframe [CC17]. An alternative approach is to first initialize a large number of keyframes and later, in the local mapping step, cull down and marginalize the redundant ones [EKC17, MAMT15]. We use this latest method, as it makes the tracking more robust to rapid motions and allows to maintain a sliding window optimization with close keyframes.

Keyframe creation is mainly associated with visual change, related to rotation and/or translation or due to lighting changes. This task is commonly addressed by setting thresholds to the following criteria: 1) a maximum rotation and translation distance, 2) a minimum number of inlier points, 3) after a fixed number of tracked frames or, 4) due to a strong change in brightness parameters.

Similar to [KSC13a], we propose the keyframe creation to be associated with the entropy reduction ΔH of the camera pose. Differently from [KSC13a], we obtain the entropy reduction independently for each degree of freedom $x \in \mathbf{x}$ using the Schur complement on the covariance matrix. We set the entropy $H^*(x_0)$ of the first frame immediately after the keyframe as the reference. This means that, in essence, our system creates a new keyframe when a certain entropy decrement is observed in at least one of the degrees of freedom of the camera.

$$H_{\Delta}^*(x, x_0) = 1 - \frac{H^*(x)}{H^*(x_0)}. \quad (2.20)$$

It may seem paradoxical, within this information framework, to establish a threshold for the process of keyframe creation. However, in contrast to other systems that define multiple and ambiguous thresholds, it is worth noting the entropy decrement allows us to use a single value that is related to tracking information. Disaggregating the information for each particular degree of freedom adds robustness and accuracy, as the aggregated information might compensate low information values in some degrees of freedom with higher ones in some others.

2.6.2 Keyframe marginalization

Keyframe marginalization is essential to keep the optimization size-bounded, enabling real-time operation [EKC17, MAT17a]. The marginalization criteria depend on whether we optimize a local map or a sliding window of keyframes. For the first case, the aim should be detecting and removing redundant keyframes, allowing lifelong operation in the same environment without unlimited growth of the number of keyframes unless the visual content of the scene changes [MAT17a]. The second technique, adopted by odometries, maintains a sliding window around the last keyframe, sufficiently spaced for an accurate optimization of the point depths.

Our marginalization belongs to the second group. However, instead of using a heuristically designed function to keep the keyframes spatially distributed, we use the mutual information measurement in order to delete the redundant ones.

Partial marginalization using the Schur complement. Instead of simply dropping out keyframes and points from the optimization, and in order to preserve most of the information,

we substitute the non-linear terms with a linearized expression of the photometric error (as in [EKC17, UESC16, VSUC18]).

The state vector update in equation (2.4) is first written in the following form

$$\begin{bmatrix} \mathbf{H}_{\alpha\alpha} & \mathbf{H}_{\alpha\beta} \\ \mathbf{H}_{\beta\alpha} & \mathbf{H}_{\beta\beta} \end{bmatrix} \begin{bmatrix} \mathbf{y}_\alpha \\ \mathbf{y}_\beta \end{bmatrix} = \begin{bmatrix} \mathbf{b}_\alpha \\ \mathbf{b}_\beta \end{bmatrix}, \quad (2.21)$$

where α and β are the blocks of variables we would like to keep and marginalize respectively. Applying the Schur complement we obtain

$$\mathbf{H}_\alpha^* = \mathbf{H}_{\alpha\alpha} - \mathbf{H}_{\alpha\beta}\mathbf{H}_{\beta\beta}^{-1}\mathbf{H}_{\beta\alpha} \quad (2.22)$$

$$\mathbf{b}_\alpha^* = \mathbf{b}_\alpha - \mathbf{H}_{\alpha\beta}\mathbf{H}_{\beta\beta}^{-1}\mathbf{b}_\beta, \quad (2.23)$$

which represents again a linear system for the state vector update, but in this case with variables β marginalized out. We can hence write a quadratic function on \mathbf{y} that can be added to the photometric error during all subsequent optimization and marginalization operations, replacing the corresponding non-linear terms:

$$r(\delta\mathbf{y}_\alpha)|_{\mathbf{y}_\alpha} = \frac{1}{2}\delta\mathbf{y}_\alpha^T\mathbf{H}_\alpha^*\delta\mathbf{y}_\alpha - \delta\mathbf{y}_\alpha^T\mathbf{b}_\alpha^*. \quad (2.24)$$

Note that partial marginalization fixes the linearization point of the variables involved, and then this would require the tangent space to remain the same over all subsequent optimization and marginalization steps. To reduce this problem we perform a relinearization of $r(\delta\mathbf{y}_\alpha)|_{\mathbf{y}_\alpha}$, as in [UESC16], every time the state is updated, i.e.,

$$\begin{aligned} & r(\delta\mathbf{y}_\alpha)|_{\mathbf{y}_\alpha+\Delta\mathbf{y}_\alpha} \\ &= r(\Delta\mathbf{y}_\alpha)|_{\mathbf{y}_\alpha} + \frac{1}{2}\delta\mathbf{y}_\alpha^T\mathbf{H}_\alpha^*\delta\mathbf{y}_\alpha - \delta\mathbf{y}_\alpha^T(\mathbf{b}_\alpha^* - \mathbf{H}_\alpha^*\Delta\mathbf{y}_\alpha). \end{aligned} \quad (2.25)$$

Similar to [EKC17], when dropping a keyframe we first marginalize all points referred to it and then the keyframe itself.

Redundancy detection using Mutual Information. As in [SC19], the redundancy $\psi(\mathcal{K}_j)$ of a keyframe with respect to the others can be expressed by

$$\psi(\mathcal{K}_j) = \sum_{i \in \mathcal{K}} MI(i, j, \Sigma_{(i,j) \setminus \mathcal{K} - \{i,j\}}), \quad (2.26)$$

where the Mutual Information between every pair of keyframes (i, j) is computed from their conditional covariance matrix $\Sigma_{(i,j) \setminus \mathcal{K} - \{i,j\}}$ with respect to the rest. This metric is used to remove, when necessary, the less informative keyframe within the window.

2.7 Experimental Results

For our evaluation we use the public TUM RGB-D benchmark [SEE⁺12a]. This dataset contains several indoor sequences, captured with an RGB-D camera and annotated with ground truth camera poses. Specifically, we use all static sequences except those beyond the range of the sensor (see Table 2.1 for the sequence list).

This section is divided into four sets of experiments. The first set evaluates the informative point selection procedure introduced in section 2.5.1. The next set analyses the keyframe creation criterion that we propose in section 2.6.1. The third set shows an analysis of computational performance. Finally, we compare our system against several state-of-the-art RGB-D odometry and SLAM systems.

The error metrics chosen for the following figures and tables are the translational keyframe-to-frame error (K2FE), used for evaluating our informative point selection, and the root-mean-square errors of translational drift in m/s (RPE) and Absolute Trajectory Error (ATEs) for comparing against state-of-the-art baselines.

		RPE (m/s)				ATE (m)		
		[KSC13a]	[MAT17a] [†]	[ZLK18]	Ours	[MAT17a] [†]	[ZLK18]	Ours
1	fr1 desk [‡]	0.024	0.051	0.031	0.029	0.065	0.044	0.051
2	fr1 floor [‡]	0.232	0.038	0.010	0.011	0.061	0.021	0.020
3	fr1 plant [‡]	0.025	0.044	0.036	0.024	0.067	0.059	0.039
4	fr1 rpy [‡]	0.032	0.037	0.034	0.026	0.066	0.047	0.045
5	fr1 xyz [‡]	0.018	0.014	0.019	0.019	0.009	0.043	0.043
6	fr2 desk	-	0.030	0.008	0.011	0.213	0.037	0.030
7	fr2 dishes	-	0.035	0.012	0.015	0.104	0.033	0.041
8	fr2 rpy	-	0.004	0.004	0.003	0.004	0.007	0.007
9	fr2 xyz	-	0.005	0.004	0.003	0.008	0.008	0.007
10	fr3 cabinet	-	0.071	0.036	0.058	0.312	0.057	0.063
11	fr3 large cabinet	-	0.100	0.167	0.049	0.154	0.317	0.096
12	fr3 long office household	-	0.019	0.010	0.010	0.276	0.085	0.038
13	fr3 nostr. text. far	0.073	0.121	0.035	0.037	0.147	0.026	0.049
14	fr3 nostr. text. near	0.028	0.050	0.043	0.015	0.111	0.090	0.062
15	fr3 str. notext. far	0.039	0.013	0.027	0.016	0.008	0.031	0.018
16	fr3 str. notext. near	0.021	0.060	-	-	0.091	-	-
17	fr3 str. text. far	0.039	0.018	0.013	0.012	0.030	0.013	0.010
18	fr3 str. text. near	0.041	0.017	0.010	0.011	0.045	0.025	0.013

Table 2.1: RMSE of translational drift RPE (m/s) and ATE (m) for state-of-the-art baselines and ID-RGBDO (Ours). Remarkably, ID-RGBDO (Ours) tracks only 24 points per keyframe. [†] stands for ORB-SLAM2-based odometry, where loop closure was deactivated from the original implementation of [MAT17a]. [‡] stands for special initialization for tracking convergence.

2.7.1 Informative point selection

We evaluate the performance of our system both quantitatively and qualitatively in terms of trajectory estimation and computational performance.

Figure 2.3 shows the translational keyframe-to-frame error (K2FE) using a number of points between 24 and 256 for all sequences we evaluated (more than 20,000 frames). The four configurations shown refer to different alternatives for point selection: completely random (**rand**), distributed in a grid and above an intensity gradient threshold (**grid**), based on our criterion to maximize the entropy of the pose (see equation (2.18)) (**inf**) and with a mixed approach between the last two (**inf+grid**). The figure shows that our information-based criterion, both combined with the grid approach and not, leads to the highest accuracy. The difference between the four alternatives is smaller as the number of points increases, but the information-based selection always results in a higher accuracy. The negligible difference in accuracy between **inf** and **inf+grid** is relevant for real-time performance, as grid-based point pre-selection is significantly faster than choosing them only based on information criteria. This is why in our

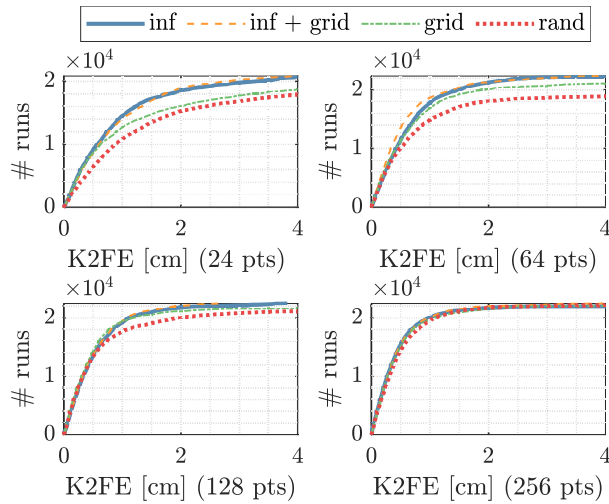


Figure 2.3: **Point Informative Selection.** Accumulated translational keyframe-to-frame error (K2FE) in all sequences. Different lines correspond to point selection modes.

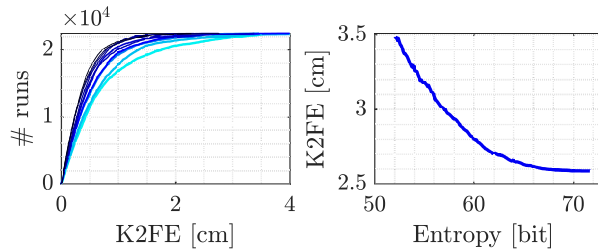


Figure 2.4: **Accuracy vs Entropy.** Left: Accumulated K2FE. Color degradation from black to blue indicates a higher entropy reduction. Right: Accumulated K2FE vs absolute entropy values.

RGB-D odometry we adopt this mixed approach.

The relation between entropy reduction and accuracy is shown in Figure 2.4. In short, the cost (the number of points needed) of improving pose accuracy increases with the absolute value of the entropy. A limitation of our current research is that, for different sequences, the specific shape of the entropy-accuracy curve is slightly different. As shown in Figure 2.5, two sequences with similar entropy values have different translational errors. These discrepancies may be due to the need of a better photometric model, as for example scenes with strong presence of motion blur give poor performance. This is not relevant for our current selection criterion, that uses relative entropy. However, future work to understand this effect could lead to further improvements.

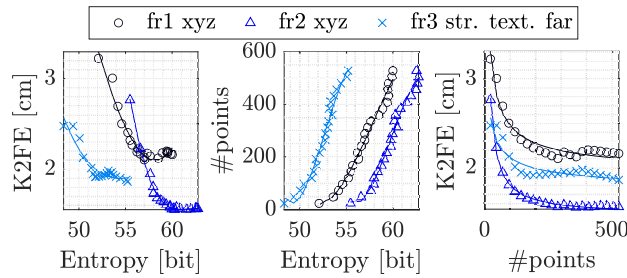


Figure 2.5: **Entropy reduction.** Left: translational keyframe-to-frame error (K2FE) vs. entropy reduction. Center: number of points vs. entropy reduction. Right: K2FE vs. number of points. The three different colors stand for three different sequences.

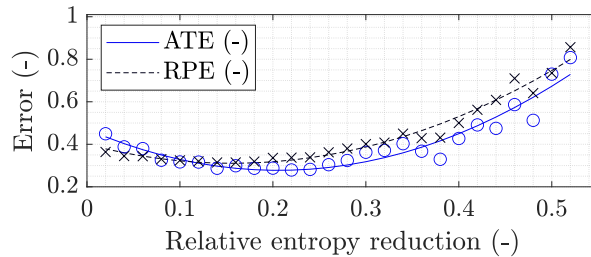


Figure 2.6: **Keyframe Creation.** Both RPE and ATE trajectory errors are influenced by the keyframe creation strategy. The figure shows a value for the relative entropy decrement H_{Δ}^* where both errors are minimal.

2.7.2 Informative Keyframe Creation

Here we demonstrate the adequacy of our entropy-based criterion for keyframe creation. Figure 2.6 shows the variation of the normalized trajectory errors (RPE and ATE), aggregated over all sequences, versus the threshold on the relative entropy reduction H_{Δ}^* to create a new keyframe. Low values lead to a high number of keyframes, which might increase the drift. Increasing the threshold on the relative entropy reduction decelerates the keyframe creation, reducing the overlap and increasing the error and eventually leading to tracking failure. Notice how this effect is modeled in the curves of Figure 2.6, and that they can be used to choose a reasonable threshold.

2.7.3 Computational Performance

We run all experiments on a laptop with an Intel Core i7-7500U CPU at 2.70 GHz and 8 GB of RAM. Figure 2.7 shows the linear dependence of the tracking cost (with and without

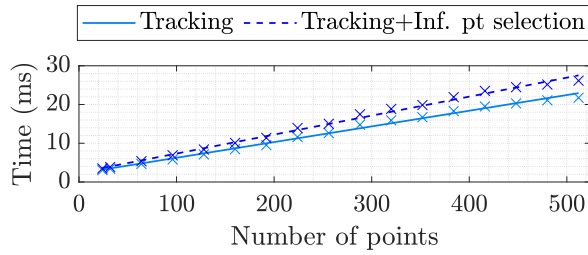


Figure 2.7: **Tracking cost**. Observe its linear growth with the number of points, and hence the convenience of using a small number of them. Observe also the small overhead introduced by our informative point selection.

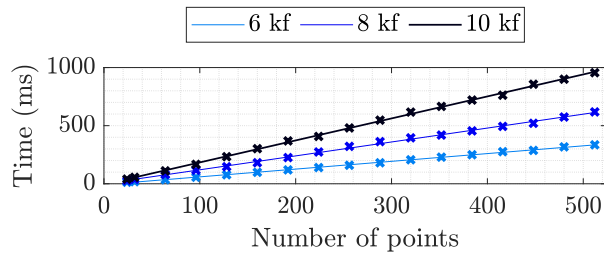


Figure 2.8: **Bundle Adjustment cost**. Notice the steep growth with the number of points, that our selection algorithm reduces with a minimal impact in the accuracy.

informative point selection) with the number of image points. Time is reduced between $5\times$ and $10\times$ from the usual practice of tracking hundreds of points to our minimal setup of 24 points. Notice also that the overhead introduced by our informative point selection algorithm is small compared to the total tracking cost, in particular for a low number of points.

Figure 2.8 shows the cost of our Direct Bundle Adjustment depending on the number of points and cameras. For our minimal configuration of 24 points per keyframe, the cost is reduced approximately $10\times$ with respect to more usual setups that optimize hundreds of points.

2.7.4 Evaluation against SotA baselines

We compare our system against three different baselines. Firstly, against Canny-VO [ZLK18], a recent RGB-D odometry based on geometric edge alignment. Secondly, against an ORB-SLAM2-based odometry, for which the original ORB-SLAM2 [MAT17a] was used with its loop closure deactivated. And, thirdly, against DVO_SLAM [KSC13a], a dense direct RGB-D SLAM. The results for ORB-SLAM2-based odometry were taken from [ZLK18]. Table 2.1 shows the

trajectory errors for these three baselines and ID-RGBDO. In our ID-RGBDO we use 24 points per keyframe and 8 keyframes in the sliding-window Bundle Adjustment. In the case of fr1, as these sequences have high motion blur due to quick rotations, we use initially a higher amount of points to aid tracking convergence but then within the Bundle Adjustment we stick to the 24 most informative points per keyframe and 8 keyframes configuration.

Notice how, for a large part of the fr2 and fr3 camera sequences, that are rich in texture and/or structure, our algorithm outperforms the three baselines. Our tracking fails in sequence 16, as all direct odometries do, while the feature-based ORB-SLAM2 succeeds. We have detected that this is due to the fact that the problem is not well conditioned with a photometric cost function and however it is conditioned enough if features are used. This result tells us how beneficial a mixed direct-features system managed with information measurements could be.

2.8 Conclusions and Future Work

In this work we have proposed a novel criterion to select the most informative points to be tracked in a RGB-D odometry framework. We have shown experimentally that using a small number of very informative points and keyframes can have a significant impact in the computational cost of RGB-D odometry, while keeping an accuracy similar to the state of the art. Specifically, our experimental results show that tracking the 24 most informative points is enough to match the performance of the state of the art while reducing the computational cost up to a factor $10\times$.

Up to our knowledge, this is the first time that information theory is applied to direct odometry and SLAM methods. We believe that our results will facilitate the use of visual odometry and SLAM in small robotic platforms and AR/VR glasses, that are limited in computation and power.

There are several lines of research that build on and could improve the results of this work. Firstly, the development of a probabilistic photometric model could improve the accuracy of the information metrics. And secondly, we also think that further analysis on the information of the windowed keyframe optimization could offer even better results. We plan to investigate both topics in the near future.

Chapter 3

■ A Model for Multi-View Residual Covariances based on Perspective Deformation

3.1 Abstract

In this work, we derive a model for the covariance of the visual residuals in multi-view SfM, odometry and SLAM setups. The core of our approach is the formulation of the residual covariances as a combination of geometric and photometric noise sources. And our key novel contribution is the derivation of a term modelling how local 2D patches suffer from perspective deformation when imaging 3D surfaces around a point. Together, these add up to an efficient and general formulation which not only improves the accuracy of both feature-based and direct methods, but can also be used to estimate more accurate measures of the state entropy and hence better founded point visibility thresholds. We validate our model with synthetic and real data and integrate it into photometric and feature-based Bundle Adjustment, improving their accuracy with a negligible overhead.

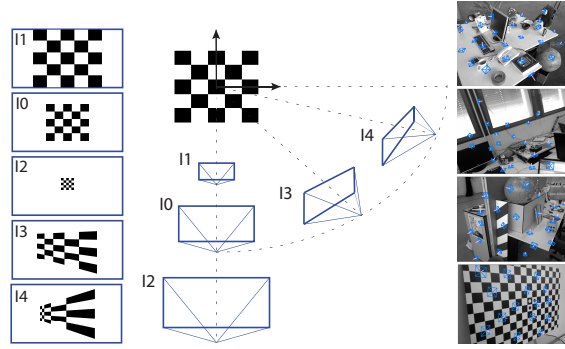


Figure 3.1: **Perspective deformation.** Image patches are subject to transformations when they are viewed from changing viewpoints, similar to the checkerboard in the image. This perspective deformation increments the covariance of the photometric/feature patches used by odometry and SLAM.

3.2 Introduction

We refer as perspective deformation to the transformations that apply to image patches when they are viewed from another viewpoint. Assuming constant camera intrinsics, it is the relative motion between a tridimensional surface and the camera poses what triggers perspective deformation in images. Figure 3.1 shows an illustrative example where such deformations can be appreciated in a checkerboard pattern. In an abuse of language, throughout the chapter we will use the terms traction and compression to characterize this perspective deformation. However, it should be remarked that we do not address deformable scenes but rigid environments.

Perspective deformation is a purely geometric effect and, yet, it is acknowledged as a challenge in many computer vision tasks. For example, SfM/odometry/SLAM pipelines, based on feature matching or photometric residuals, iterate over several pyramid levels [EKC17] or set heuristic thresholds reflecting low confidence for wide-baseline matches [MAT17a]. The accuracy of a camera calibration can be modelled as the trade-off between a sufficiently informative geometric configuration and the image noise that perspective deformations produce [PS19]. In other tasks such as semantic segmentation or object/place recognition, perspective deformation is also an issue if viewpoints vary significantly [SVN20, HCMH16, GSM19].

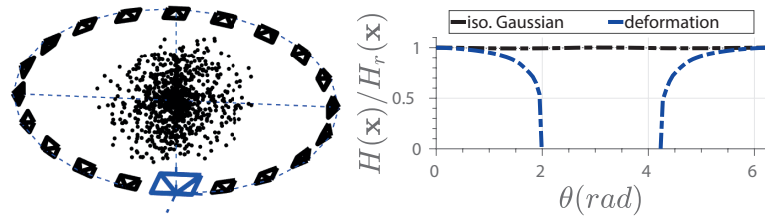
For the specific case of multi-view reconstruction, when looking at Figure 3.1, it is evident that a high degree of perspective deformations will also distort appearance-based descriptors, resulting in noisier image matches. However, visual residuals \mathbf{r} are modeled as isotropic Gaussians $\mathbf{r} \sim \mathcal{N}(\mathbf{0}, \Sigma_r)$, $\Sigma_r = \sigma_r^2 \mathbf{I}$ in the vast majority of 3D vision pipelines. The visual residual model

has a direct influence in the accuracy of the camera and structure states \mathbf{x} via the Gauss-Newton updates $\Delta\mathbf{x} = -(\mathbf{J}^\top \boldsymbol{\Sigma}_r \mathbf{J})^{-1} \mathbf{J}^\top \mathbf{r}$ (\mathbf{J} stands here for the derivatives of the residuals \mathbf{r} with respect to \mathbf{x}). In this chapter we propose a new model for the covariances of the visual residuals σ_r^2 that accounts for the effect of the perspective deformation and hence improves the accuracy of multi-view structure and motion estimations.

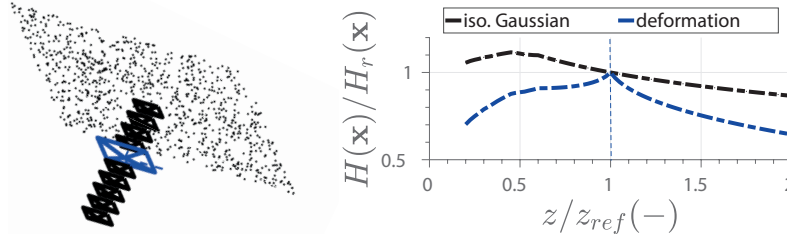
Furthermore, as [PS19] points out, in many practical applications one should incorporate estimates of the uncertainty when available. The covariances $\boldsymbol{\Sigma}_x$ over the state \mathbf{x} are usually back-propagated from the residual covariance $\boldsymbol{\Sigma}_r$ as $\boldsymbol{\Sigma}_x = (\mathbf{J}^\top \boldsymbol{\Sigma}_r^{-1} \mathbf{J})^{-1}$ [SMD10]. A better model for the residual covariances $\boldsymbol{\Sigma}_r$ would lead to more realistic uncertainty estimates, which is crucial in real-world applications.

As a final application case, using information metrics in odometry and SLAM dates back to works such as [SMD10, CD08, KSC13a] but has seen great progress recently —aiming to the reduction of the computational demand for their implementation on low-end platforms [FCT20, KMZS20, ZV18, ZV20]. Again, a better model for $\boldsymbol{\Sigma}_r$ would significantly improve such approaches. As two illustrative examples, Figure 3.2 shows inconsistencies that arise when the differential entropy $H(x) = -\frac{1}{2} \log((2\pi e)^k |\boldsymbol{\Sigma}_x|)$ is obtained approximating the residual distribution by an isotropic Gaussian.

As a summary, the specific contributions of this chapter are as follows. First, we derive a model for the perspective deformation of 2D image patches (Section 3.4). To the best of our knowledge, we are the first ones addressing such deformation in a general manner. Second, we introduce a model for the visual residuals based on the perspective deformation, valid for both feature-based and photometric methods (Section 3.5). Third, we validate our model with extensive experimentation in a realistic synthetic dataset and real data (Section 3.6). To our knowledge, this is the first time that the relation between perspective deformation and multi-view residuals is shown and characterized in several setups under a unified derivation. Finally, we integrate our model in the global optimization of feature-based and direct odometry/SLAM pipelines, demonstrating a consistent reduction of the trajectory error in the TUM RGB-D dataset [SEE⁺12a] (Section 3.7).



(a) **Circular trajectory.** For isotropic Gaussian residuals, the differential entropy is incorrectly modeled as constant even for 180° parallax. Our deformation-based covariance models it correctly, showing a steep decrease with parallax.



(b) **Approaching trajectory.** For isotropic Gaussian residuals, the differential entropy increases as the camera approaches the planar scene, which is not correct. With our deformation-based model areas close to the reference frame (the blue one) give the best accuracy.

Figure 3.2: Differential entropy inconsistencies. A deficient model of the residuals leads to inconsistencies in a variety of applications. An illustrative one is the camera pose entropy in these two situations.

3.3 Related Work

Isotropic visual residuals are widespread in multi-view setups [KM07, EHS⁺13, WS14, SF16, MAT17a, EKC17, RACC20] and only a few exceptions differ or are directly related to our work. The work of [YWGC18] implicitly underlines the importance of visual covariances by evaluating aspects such as photometric calibration, motion bias and rolling shutter, and their effect on direct, feature-based, and semi-direct odometries.

Molton et al. [MDR04] models salient features as observations of locally planar regions, compensating for the predicted motion before matching. Such early model is, however, limited to template matching based on cross-correlation. More recently, for the application of camera calibration, Peng and Sturm [PS19] incorporate uncertainty for the corners of a calibration target using autocorrelation matrices.

Engel et al. [EKC17] apply a gradient-dependent weighting, reducing the effect of photometric errors in pixels with high gradient. This can be probabilistically explained as approximating the geometric error by adding on the projected point position, small and independent geometric noise, and directly marginalizing it. Mur-Artal and Tardós [MAT17a] scale the visual residual

proportionally to the resolution where the ORB features are detected. In both works, apart from these two aspects, the noise model follows the standard isotropic Gaussian assumption. Up to our knowledge, ours is the first model deriving a probabilistic form of perspective deformation, opening a research line towards a better understanding and a general modeling of visual residuals.

3.4 Perspective Deformation

3.4.1 Preliminaries

We refer with subscript j to the reference frame where a 3D point $\mathbf{p} \in \mathbb{R}^3$ is first observed, and with i to any other frame from which the point is visible. The image coordinates of the projection of \mathbf{p} in reference frame j and its depth are denoted as $\mathbf{u} \in \Omega$ and $z \in \mathbb{R}$ respectively, where Ω is the image domain.

The function $\varphi(\mathbf{u})$ projects a point \mathbf{p} from its camera coordinates \mathbf{u} in the reference frame j into the frame i ,

$$\varphi(\mathbf{u}) = \Pi(\mathbf{R}\Pi^{-1}(\mathbf{u}, z) + \mathbf{t}), \quad (3.1)$$

where $\Pi(\mathbf{p})$ (determined by the intrinsic camera parameters) projects the point \mathbf{p} in the camera frame; and $\Pi^{-1}(\mathbf{u}, z)$ back-projects the image point with coordinates \mathbf{u} at depth z . $\mathbf{R} \in \text{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$ are the relative rotation and translation between frame j and frame i .

3.4.2 Surface representation

We consider that each point \mathbf{p} lays on a local 3D surface S . Our formulation can include surfaces with any degree of complexity as far as the depth z of \mathbf{p} can be expressed as a function of its image coordinates $z = S(\mathbf{u})$. Similarly to [MDR04], in our implementation we constrain those surfaces to be 3D planes $S = f(\alpha, \beta, \gamma)$; with α , β and γ being the plane parameters on the camera reference frame. We then can formulate the depth z for each point \mathbf{p} in terms of its camera coordinates \mathbf{u} and its corresponding local plane parameters (α, β, γ) :

$$z = \frac{\gamma}{1 - [\alpha, \beta, 0] \cdot \Pi^{-1}(\mathbf{u}, 1)}. \quad (3.2)$$

Commonly, direct VO and SLAM extend point descriptors over a small neighborhood of pixels [EKC17]. Feature-based pipelines [SF16, MAT17a] with classic feature descriptors [Low04, RRKB11] also perform operations on patterns around a central pixel. Operating on a pattern of pixels on the image is equivalent to consider that all these pixels have the same local depth coordinate ($\alpha = 0, \beta = 0, \gamma = z$). Note that our former assumption is a compromise between considering more complex surfaces and assuming that the point belongs to a plane orthogonal to the local z-axis.

3.4.3 Perspective deformation model

We will approximate the projection function of Equation (3.1) by its **first-order Taylor approximation**

$$\varphi(\mathbf{u} + d\mathbf{u}) \approx \varphi(\mathbf{u}) + \nabla_{\mathbf{u}}\varphi d\mathbf{u}. \quad (3.3)$$

The **perspective deformation gradient tensor** $\mathbf{F}_{\psi}(\mathbf{u})$ contains all the information about the local rotation ψ and deformation of \mathbf{u} and corresponds to the Jacobian matrix of the transformation $\varphi(\mathbf{u})$,

$$\mathbf{F}_{\psi}(\mathbf{u}) = \nabla_{\mathbf{u}}\varphi = \frac{\partial\varphi(\mathbf{u})}{\partial\mathbf{u}} = \left[\frac{\partial\mathbf{u}}{\partial\mathbf{p}_n} \frac{\partial\mathbf{p}_n}{\partial\mathbf{p}_c} \frac{\partial\mathbf{p}_c}{\partial\mathbf{p}_w} \right]_i \left[\frac{\partial\mathbf{p}_w}{\partial\mathbf{p}_c} \frac{\partial\mathbf{p}_c}{\partial\mathbf{p}_n} \frac{\partial\mathbf{p}_n}{\partial\mathbf{u}} \right]_j, \quad (3.4)$$

where subscripts n , c and w correspond to the point coordinates normalized, in the camera frame and in the absolute one respectively. Note that the gradient tensor $\mathbf{F}_{\psi}(\mathbf{u})$ models how an infinitesimal line segment in the “undeformed” reference frame is not only stretched but also rotated with an angle ψ into a line segment in the “deformed” frame.

The **Cauchy–Green deformation tensor** gives a measure of the deformation that is independent of the rotation around the camera axis, without needing explicitly the rotation matrix. By applying the polar decomposition theorem, which states that any second-order tensor can be

decomposed into a product of a pure rotation and symmetric tensor, it is possible to separate the camera rotation \mathbf{R}_ψ from a rotation-independent deformation gradient tensor \mathbf{F}_u , hence $\mathbf{F}_\psi(\mathbf{u}) = \mathbf{R}_\psi \mathbf{F}_u = \bar{\mathbf{F}}_u \mathbf{R}_\psi$.

The tensor \mathbf{C} is called the right Cauchy–Green deformation tensor

$$\mathbf{C} = \mathbf{F}_\psi(\mathbf{u})^T \mathbf{F}_\psi(\mathbf{u}) = \mathbf{F}_u^T \mathbf{R}_\psi^T \mathbf{R}_\psi \mathbf{F}_u = \mathbf{F}_u^T \mathbf{F}_u. \quad (3.5)$$

Since it is formed only from the \mathbf{F}_u tensor, it describes the deformation of the material “before” rotation.

The left Cauchy–Green deformation tensor $\bar{\mathbf{C}}$

$$\bar{\mathbf{C}} = \mathbf{F}_\psi(\mathbf{u}) \mathbf{F}_\psi(\mathbf{u})^T = \bar{\mathbf{F}}_u \mathbf{R}_\psi \mathbf{R}_\psi^T \bar{\mathbf{F}}_u^T = \bar{\mathbf{F}}_u \bar{\mathbf{F}}_u^T, \quad (3.6)$$

applies a rigid body rotation first, and then deforms the rotated volume. Both tensors are independent of the rotation, but they describe the deformation in different frames.

Deformation. Physically, the Cauchy–Green tensor gives us the square of the local geometric changes $\varepsilon^2(\boldsymbol{\eta})$ due to deformation in some particular directions $\boldsymbol{\eta}$:

$$\varepsilon^2(\boldsymbol{\eta}) = \boldsymbol{\eta}^T \mathbf{C}(\mathbf{u}) \boldsymbol{\eta}. \quad (3.7)$$

If we consider a single direction of interest $\boldsymbol{\eta} \in \mathbb{R}^2$ in which we want to obtain the perspective deformation (*e.g.*, the direction of the gradient for photometric errors), we then obtain a scalar deformation $\varepsilon^2 \in \mathbb{R}$. On the other hand, if there are two directions of interest $\boldsymbol{\eta} \in \mathbb{R}^{2 \times 2}$ (*e.g.*, the geometric residuals in feature-based methods) the deformation obtained is not only 2-dimensional but also anisotropic $\varepsilon^2 \in \mathbb{R}^{2 \times 2}$.

Traction and Compression are relative terms that depend on which of the configurations we consider as “undeformed”. Due to the linearized projection model (equations (3.3) and (3.4)), we can derive that the inverse transformation $\mathbf{F}^{-1}(\mathbf{u})$ yields the inverse stretch $|\mathbf{F}^{-1}(\mathbf{u})| = |\mathbf{F}(\mathbf{u})|^{-1}$. In other words, we can map every compression $\varepsilon_c^2 \in [0, 1)$ into its homologous traction $\varepsilon_t^2 \in (1, \infty)$ and viceversa just by $\varepsilon_t^2 \approx \varepsilon_c^{-2}$.

3.5 Visual Residual Covariances

Deformation Covariance. Under traction ($\varepsilon^2 > 1$), the covariances of the visual residuals grow as a function of the deformation according to certain response functions $\sigma_t^2 : \varepsilon^2 \subset \mathbb{R} \mapsto \mathbb{R}$. These functions vary with the residual model used for each particular application (e.g., feature-based or photometric) and we determine them experimentally in this work (see the validation in Section 3.6 and further experiments in Section 3.7). As mentioned, we will model an equivalent traction for every scalar compression with the response function $\sigma_c^2 : \varepsilon^{-2} \subset \mathbb{R} \mapsto \mathbb{R}$. To sum up, for the case $\boldsymbol{\eta} \in \mathbb{R}^2$ our model results in

$$\sigma_\varepsilon^2(\varepsilon^2) = \begin{cases} \sigma_c^2(\varepsilon^{-2} - 1) & \varepsilon^2 \leq 1 \\ \sigma_t^2(\varepsilon^2 - 1) & \varepsilon^2 > 1. \end{cases} \quad (3.8)$$

With this formulation we aim for compression and traction to have similar response functions that map deformations into visual covariances ($\sigma_t^2 = \sigma_c^2$). However, effects such as pixel discretization or processing done by feature-based approaches induce more complex behaviours for these response functions. We propose and analyze some particular cases in our validation experiments in Section 3.6.

2d-deformation. If residual covariances are coupled in two image directions $\boldsymbol{\sigma}_\varepsilon^2 \in \mathbb{R}^{2 \times 2}$ (such as in corner matching), $\mathbf{C} \in \mathbb{R}^{2 \times 2}$ is a diagonalizable symmetric positive semi-definite matrix. Then, it can be found a unitary matrix $\mathbf{V} \in \mathbb{R}^{2 \times 2}$ where the matrix containing the deformation (3.7) in each direction $\boldsymbol{\varepsilon}_{kk}^2 = \mathbf{V}\mathbf{C}\mathbf{V}^T \in \text{diag}(\mathbb{R}^2)$ is diagonal. We obtain the non-diagonal covariance matrix applying the model in Equation (3.8) to the diagonal elements of $\boldsymbol{\varepsilon}_{kk}^2$ and undoing the transformation $\boldsymbol{\sigma}_\varepsilon^2 = \mathbf{V}\boldsymbol{\sigma}_\varepsilon^2(\boldsymbol{\varepsilon}_{kk}^2)\mathbf{V}^T \in \mathbb{R}^{2 \times 2}$.

Projection Covariance. In addition to perspective deformation, there are other possible noise sources (e.g., rolling shutter effects [CZK15, YWGC18]) that are propagated through the projection function to the visual residuals and can be added as geometric uncertainties in our covariance $\boldsymbol{\sigma}_\varphi^2$.

As an example, the depth uncertainty from stereo cameras and RGB-D ones using structured light can be propagated from the disparity variance σ_v^2 . Assuming a focal length f , a baseline

b and a disparity ν , the first-order propagation for the inverse depth covariance is [HWMD14, Kho11, BM13, CC17]

$$z = \frac{fb}{\nu}, \quad \sigma_z = \frac{fb}{\nu^2} \sigma_\nu = \frac{z^2}{fb} \sigma_\nu. \quad (3.9)$$

Using a first-order propagation of the projection in Equation (3.1), we obtain the contribution of the depth uncertainty to the residual

$$\sigma^2(z) = \left(\frac{\partial \mathbf{u}}{\partial z} \right)^2 \sigma_z^2. \quad (3.10)$$

Finally, all uncertainty contributions can be grouped together into a single term that models the full covariance of a visual measure in a given direction

$$\sigma_\varphi^2 = \sigma_\varepsilon^2(\boldsymbol{\eta}) + \boldsymbol{\eta}^T (\sigma^2(z) + \dots) \boldsymbol{\eta}. \quad (3.11)$$

3.5.1 Implementation Details

For the sake of reproducibility, we describe several practical aspects of the implementation of our model, namely point visibility, photometric errors and feature matching.

Photometric residual. Direct methods define a photometric error between the raw image intensities $I : \Omega \subset \mathbb{R}^2 \mapsto \mathbb{R}$. Although each method has specific particularities in their residual definitions, most of them concur in evaluating the photometric error $r \in \mathbb{R}$ in a slightly spread pattern of pixels \mathcal{P} [EKC17, FPS14]

$$r = \sum_{u \in \mathcal{P}} (I_j(\mathbf{u}) - I_i(\varphi(\mathbf{u})))^2. \quad (3.12)$$

The residual covariance σ_r^2 can be modeled in this case as a purely photometric addend σ_N^2 and the geometric covariance from Equation (3.11) propagated with the intensity gradient G

$$\sigma_r^2 = \sigma_N^2 + G^2 \sigma_\varphi^2(\boldsymbol{\eta}_g). \quad (3.13)$$

As shown in [EUC16, EKC17, YWGC18], a conscientious photometric calibration improves the accuracy and robustness of direct methods. It would be reasonable to include the model of the photometric contribution σ_N^2 at this point in the formulation. However, due to the scope of this work, we include the propagation of the image noise obtained through bilinear interpolation σ_I^2 to the photometric error calculated on a N-sized pattern (3.12)

$$\sigma_N^2 = \frac{128N}{81}(\sigma_I^2)^2. \quad (3.14)$$

If depth information is available, the pixels $\mathbf{u} \in \mathcal{P}$ could be considered to belong to the same 3D surface (similar to Equation (3.2)) and reproject them accordingly. However, monocular setups estimate depth from multiple views, making the assumption that all pixels on a pattern share the same depth quite convenient. Note how our formulation can easily incorporate this assumption with $\mathbf{u} \in S_0(\alpha = 0, \beta = 0, \gamma = z)$.

Feature-based errors are usually defined as variations of the following expression:

$$\mathbf{r} = \mathbf{u}_i - \varphi(\mathbf{u}), \quad (3.15)$$

where \mathbf{u}_i stands for the feature point in image i and $\varphi(\mathbf{u})$ for the corresponding point in the frame j reprojected in the frame i . Hence the residual covariance is expressed as the sum of the projection covariance and the feature subpixel noise σ_u^2 :

$$\sigma_r^2 = \sigma_u^2 + \sigma_\varphi^2 \quad (3.16)$$

Right/left Cauchy-Green deformation tensor. The right tensor \mathbf{C} models the deformation “before” rotation, that means in the original frame. Since direct methods compute photometric gradients in these reference frame, it is not necessary to recompute them again. On the other hand, feature-based approaches set geometric residuals in the reprojection frame, that means “after” rotation. Then by using the left Cauchy-Green tensor $\bar{\mathbf{C}}$ deformations are conveniently referred to that coordinate frame.

Point visibility. Heuristic thresholds for point visibility are a trade-off between the potential benefits of wide baselines and the increasing matching uncertainties. Some approaches keep

observations in a bunch of close keyframes and remove outliers with robust cost norms [EKC17]. Others define angle and scale thresholds between viewing rays [MAT17a]. Defining a threshold in terms of deformation is more principled, since it accounts for the relative orientation between the camera and the local surface. If a point is observed with a parallax angle bigger than 90° , the diagonal of the deformation tensor (3.4) triggers a negative value. In addition, weighting visual residuals in terms of the perspective deformation allows a wider range of inliers without degrading the optimization.

3.6 Model Validation

First, we test the basis of our model equations (Section 3.4) with a Monte Carlo-based experiment. Next, to identify the applicable cases of the perspective deformation, and explore the function-revealing parameters of Equation (3.8) that could guide good visual covariance modeling, we validate our approach in a two-branch experiment: photometric and feature-based. The estimates of the model parameters are then used as the input to the experiments in Section 3.7. Finally, we show the differences of taking into account perspective deformation in a simple application that computes the amount of available information for the tracking of a camera from a cloud of points.

3.6.1 Geometric covariance

Figure 3.3 gives an overview of the simulated setup we use to assess the deformation model. We generate a set of random cameras, points and surfaces to produce a massive number of projection samples ($\approx 10^6$). To simulate the perspective deformation suffered by planar patches, we add a small Gaussian noise to the reference coordinates of the points \mathbf{u} , and measure the covariance of the projected error distribution. Then, for each projection, we obtain a covariance matrix $\bar{\mathbf{C}}_{sim}$ analogous to the left Cauchy-Green deformation tensor in Equation (3.6). Figure 3.3f shows the relation between the simulated 2d deformation $\varepsilon_{sim}^2 = \det(\bar{\mathbf{C}}_{sim})$, and our estimation with the derivation in Section 3.4.

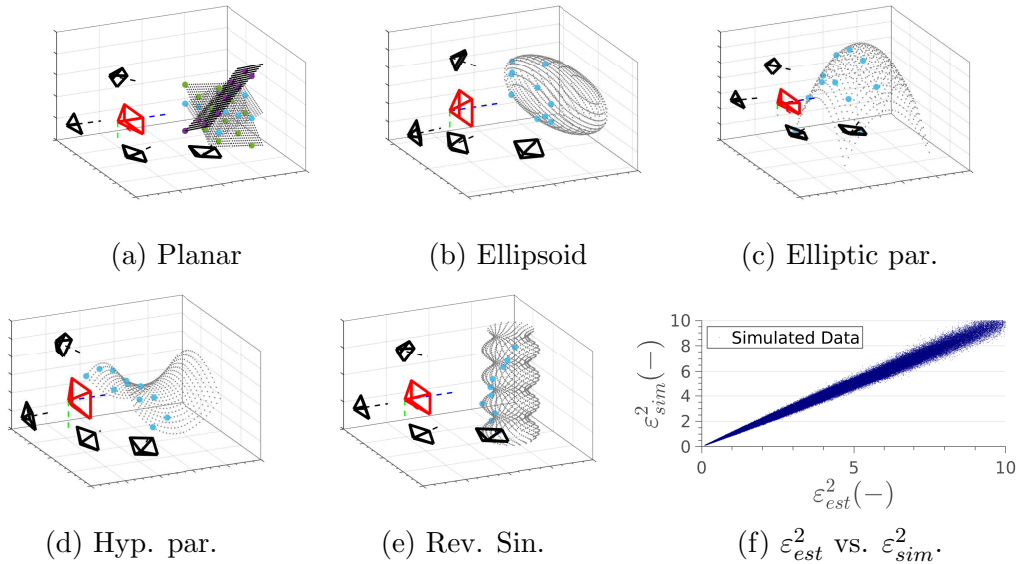


Figure 3.3: Monte Carlo validation. We show the comparison of our deformation estimation ε_{est}^2 with a simulation ε_{sim}^2 on a set of representative surfaces: planar, ellipsoid, elliptic and hyperbolic paraboloid, and a revolution sine. Figure 3.3f demonstrates how our model can be used, not only with planes, but with any parameterizable surface $z = S(\mathbf{u})$ (see Section 3.4.2).[†] Cameras and points shown in this figure are just a subset chosen with visualization purposes. The total amount of point projections is $\approx 10^6$.

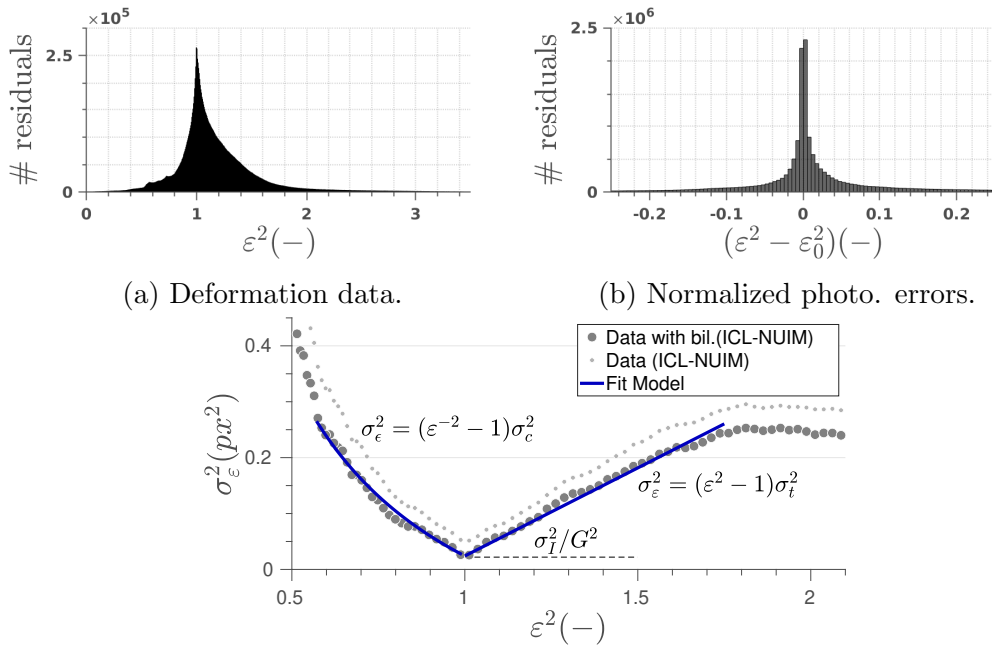
3.6.2 Photometric patches

To ensure that only the influence of perspective deformation is being considered ($\sigma_\varphi^2 = \sigma_\varepsilon^2$), we validate our model for planar photometric patches in the synthetic ICL-NUIM dataset [HWMD14], which provides RGB-D sequences and the ground truth values for the camera and scene parameters.

As in [MDR04], we consider patches as locally planar regions on 3D world surface instead as 2D templates in image space or more complex surfaces, since our formulation allows patches to be placed on any parametric surface (see Section 3.6.1). Although salient points often appear at discontinuities, it is commonly assumed that it is possible to find a locally dominant plane for their representation (see Figure 3.1).

We gather a massive number of high gradient pixel projections between all image pairs within the same sequence, and extract their photometric errors, intensity gradients and predicted deformations. Figures 3.4a and 3.4b show histograms illustrating the number of data points, the deformation range and photometric errors in the dataset.

We group photometric errors according to their deformation and normalize them with the



(c) **Photometric covariance model.** Results from the validation in the ICL-NUIM dataset [HWMD14] show that the covariance for photometric patches increases along with the perspective deformation according to Equation (3.8).

Figure 3.4: **Photometric model validation.**

intensity gradient. Finally, we compute the covariance of the errors within each cluster obtaining the value for σ_x^2 in Equation (3.13)

$$\sigma_x^2 = \frac{\sigma_r^2}{G^2} = \frac{\sigma_N^2}{G^2} + \sigma_\varepsilon^2(\varepsilon^2). \quad (3.17)$$

Figure 3.4c shows a representative sample of the results (using a pattern of 9 pixels), clearly confirming a relation between perspective deformation and visual covariances and how our model captures it accurately. We highlight three significant outcomes: **1)** expressing compressions as its homologous tractions ($\varepsilon_t^2 = \varepsilon_c^{-2}$, see Section 3.4) unifies the behaviour of both covariance responses ($\sigma_t^2 \sim \sigma_c^2$). **2)** Following Ockham’s razor, we define $\sigma_t^2(\varepsilon^2 - 1)$ and $\sigma_c^2(\varepsilon^{-2} - 1)$ simply with constant values. **3)** From the minimum covariance value in the absence of perspective deformation ($\sigma_\varepsilon^2 = 1$) we can derive with Equation (3.14) the photometric noise in the images σ_I^2 . Note how as expected the use of bilinear interpolation for intensities reduces the image noise to around $\frac{4}{9}$.

Patch patterns. Table 3.1 evaluates our model for photometric residuals in different patch patterns, as proposed in [EKC17]. The most relevant outcomes are: **1)** Bigger patches act

radius	$S\{\mathbb{R}^2 > 0.9975\}$			
	σ_I	σ_t^2	σ_c^2	ε^2
0.5	4.02	0.30	0.49	0.50-1.75
1	2.69	0.31	0.43	0.48-1.70
$\sqrt{2}$	2.66	0.32	0.40	0.46-1.66
2	2.25	0.33	0.35	0.43-1.61
$2\sqrt{2}$	1.99	0.35	0.31	0.39-1.52
4	1.74	0.36	0.28	0.33-1.41
$4\sqrt{2}$	1.64	0.38	0.22	0.25-1.25

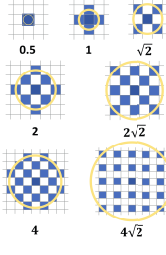


Table 3.1: **Photometric model fitting.** For each patch size, we obtain the parameters of Equation (3.8) that maximize the coefficient of determination \mathbb{R}^2 within a certain range of deformation ε^2 . Per-column best is displayed in green, worst in red and balanced in blue. Small patches behave better for traction, large patches for compression.

as photometric filters, reducing the image noise σ_I . **2)** Bigger patches experience a faster degradation of the performance under traction, but perform better than smaller patches under compression. And vice versa, smaller patches deteriorate faster under compression and handle traction better. This effect is easily recognizable observing the changes of the response function parameters (σ_t^2 , σ_c^2) or the range of deformation ε^2 in Table 3.1 .

The last columns of Table 3.1 show results assuming the surface is perpendicular to the optical axis (S_0) or the backprojected ray (S_\perp). We observe that now traction effects are mostly dominated by the **surface assumption**, *i.e.*, as the camera moves towards the points, depth inconsistencies arise. Yet, bigger patches still work better under compression.

camera	σ_I	σ_t^2	σ_c^2	$\varepsilon^2 \in$	\mathbb{R}^2	$\frac{\sigma_t^2}{\sigma_c^2}$
fr1	47.2	1.80	1.34	0.6-1.4	0.95	1.3
fr2	22.0	0.88	0.89	0.6-1.8	0.97	1.0
fr3	34.4	0.79	0.90	0.7-1.4	0.98	0.9
realSense	30.8	0.63	0.62	0.6-1.9	0.97	1.0

Table 3.2: **Model validation for photometric 9-pixel patches with real data.** The table shows the parameters of Equation (3.8) estimated for the cameras of the RGBD-TUM dataset (fr) and a realsense D435i depth camera. Note how the 9-pixel patch behaves similarly under traction and compression deformation.

Real Data. We repeat the experiment in Figure 3.4c with real data from three different cameras of the public RGB-D TUM dataset [SEE⁺12a]. Figure 3.5 shows how again, our model accurately captures the visual covariance produced by perspective deformation of photometric planar patches. An important outcome from the real data with different cameras and sequences is the validation of the assumption of considering patches as locally planar regions rather than more complex surfaces. However, it also leads to saturation of the model under strong changes

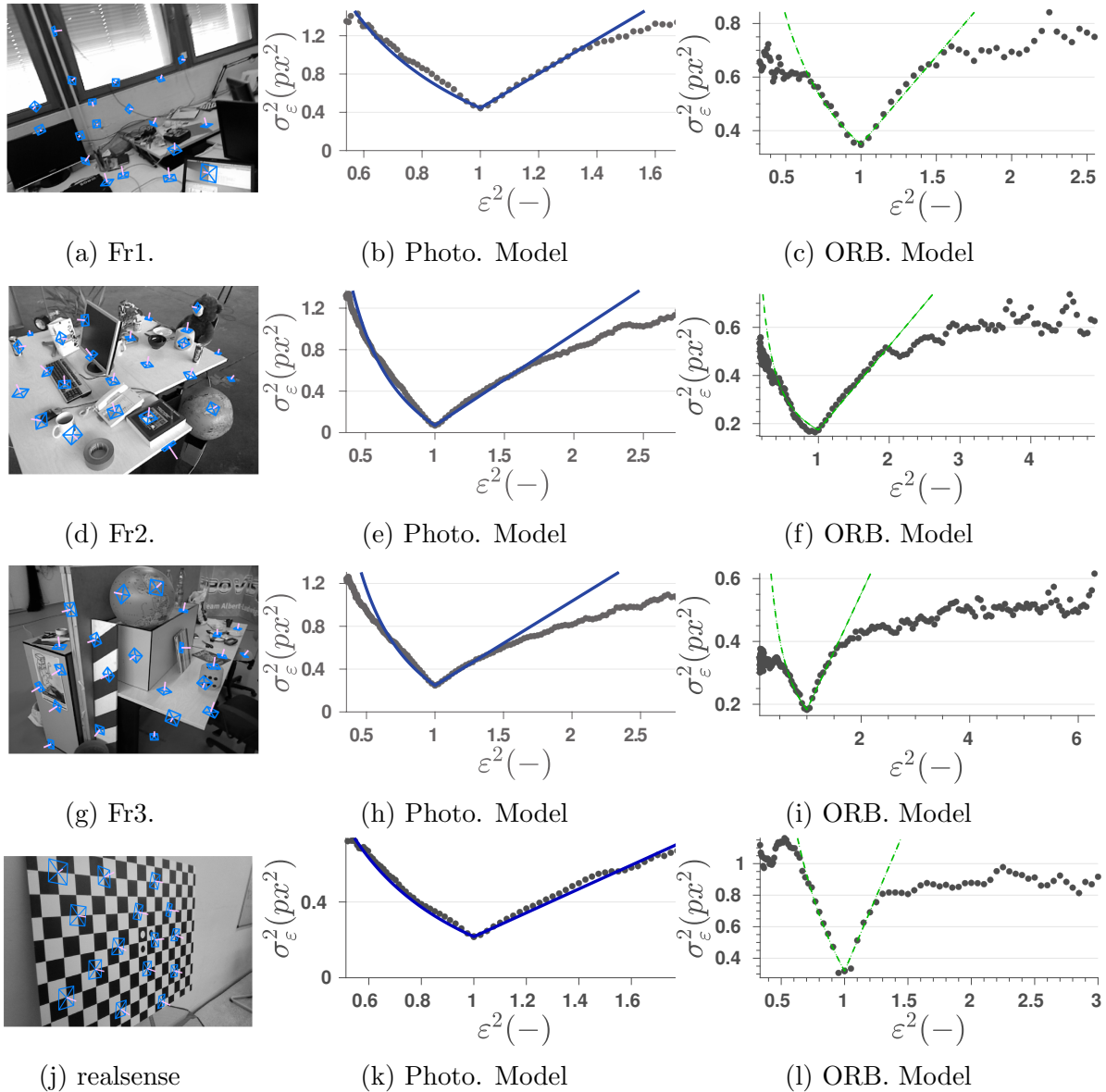


Figure 3.5: **Model validation with real data.** We perform the model fitting of Equation (3.8), described in sections 3.6.2 and 3.6.3, with data from three cameras of the sequences of the RGB-D TUM dataset [SEE⁺12a] and with data recorded with a realsense D435i depth camera. It can be seen how our covariance model for photometric patches and ORB features fits the noise distribution of the four cameras.

in perspective deformation. Finally, to have a cleaner experiment with data from a depth sensor but trying to minimize any source of noise, we validated the model with data from an intel realsense D435i depth camera facing a checkerboard pattern. Figure 3.5k shows how our model fits more reliably in this validation performed under more suitable conditions. Table 3.2 collects the coefficients of the model validation that we will use in the experiments of Section 3.7.

3.6.3 Feature-based methods

We keep the experimental setup of the previous section, but we now evaluate the geometric reprojection residual in feature-based methods. Similarly to Figure 3.4c, we report in Figure 3.6a the dependency of the reprojection error with perspective deformation for different point features. The results show again a clear relation between the perspective deformation and the visual residual, and how our model fits reasonably the simulation data.

Table 3.3 shows the results for our model fitting. In the column titled σ_t^2/σ_c^2 , it is relevant to note that visual covariances tend to grow faster for traction than compression. This is consistent with our photometric validation (see Table 3.1), where covariances in large patches grew faster under traction. Moreover, these results agree with [YWGC18], that shows experimentally the effect of motion bias in ORB-SLAM2 [MAT17a] and DSO [EKC17]. They show a noticeable degradation for ORB-SLAM2 when the camera is moving forward, meaning that points mainly approach and consequently patches suffer from traction. On the other hand, DSO using photometric patches of radius 2 (in our Table 3.1, with balanced traction and compression coefficients) does not show such bias. Our findings here are a step forward towards a more complete understanding of motion bias in VO/SLAM.

As one limitation of these results, features extracted with different filtering parameters and image resolutions introduce a scaling factor between the residual covariances and perspective deformation. So far, we extracted features at the original image resolution. The ORB implementation [RRKB11] operates at discrete scale levels s since it performs the same operations at different image resolutions. Figure 3.6b shows the dependency of the residual covariance with the perspective deformation for each of these resolutions. For our model to be used at different scales, we approximate this effect by scaling our perspective deformation covariance, where s and s_r stands for the resolution factor of the reference and projected image respectively.

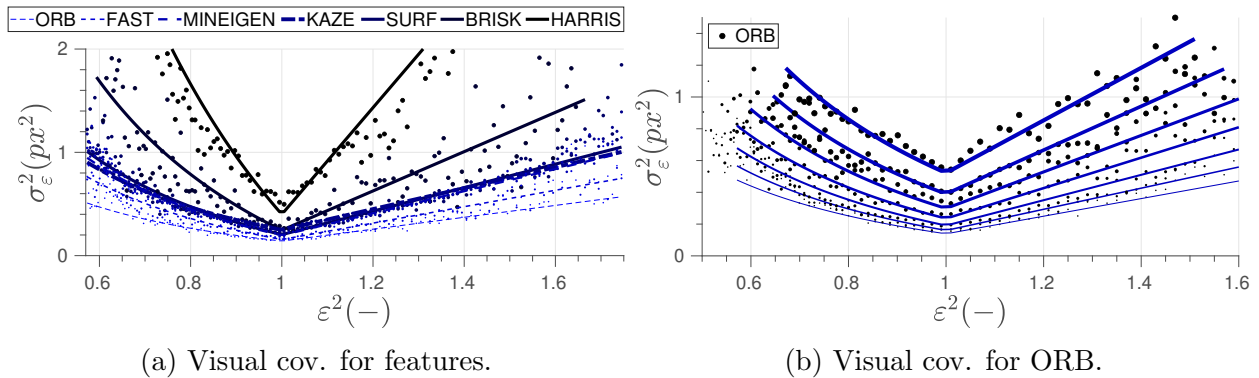


Figure 3.6: In 3.6a all descriptors increase their covariance with the deformation. 3.6b shows the deformation covariance of ORB depending of the scale of the feature. From bottom to top the resolution becomes coarser.

Finally, we repeat the validation with real data the same manner as in Section 3.6.2. Figure 3.5 and Table 3.3 show the results of the validation.

camera	σ_p	σ_t^2	σ_c^2	$\varepsilon^2 \in$	R^2	$\frac{\sigma_t^2}{\sigma_c^2}$
fr1	0.59	0.66	0.48	0.6-1.7	0.97	1.38
fr2	0.42	0.35	0.15	0.5-2.1	0.96	2.33
fr3	0.43	0.37	0.22	0.6-1.6	0.97	1.68
realSense	0.56	1.88	1.42	0.8-1.2	0.87	1.32

Table 3.3: **Model validation for ORB features with real data.** The table shows the parameters of Equation (3.8) estimated for the cameras of the RGBD-TUM dataset (fr) and a realsense D435i depth camera. Note how the visual covariance of ORB grows more in traction than in compression.

3.7 Experiments

The validation analysis in Section 3.6 showed the relation between residual covariances and perspective deformation. In this section we demonstrate its applicability in state-of-the-art pipelines. Specifically, we evaluate the accuracy improvement in the photometric Bundle Adjustment (BA) of [FCT20] and in the feature-based BA of ORB-SLAM [MAT17a].

For our evaluation we use the public TUM RGB-D benchmark [SEE⁺12a], that contains several indoor sequences captured with a RGB-D camera annotated with ground truth camera poses. Specifically, we use all static sequences except those beyond the range of the sensor. All the experiments were run on a standard laptop with an Intel Core i7-7500U CPU at 2.70 GHz and 8 GB of RAM for which the overhead caused by our model was less than 2% of the total cost.

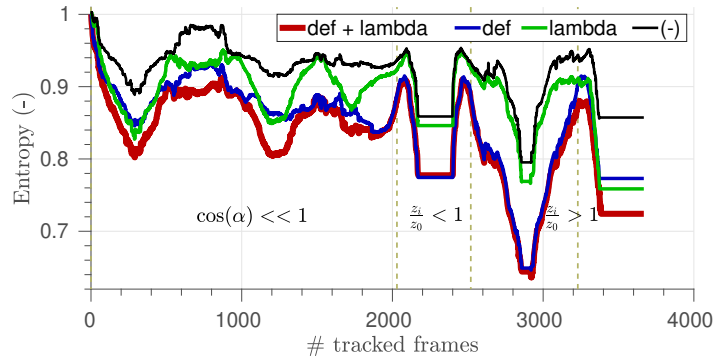


Figure 3.7: Tracking entropy $H(\text{bits})$. Big parallax ($\cos(\alpha) \ll 1$). Approximation ($\frac{z_i}{z_0} < 1$). Distancing ($\frac{z_i}{z_0} > 1$). Model complete (Def + lambda), just deformation covariance (def), just depth covariance (lambda), constant covariance (-).

3.7.1 Information Metrics

As we anticipated, deriving the differential entropy of the camera pose $H(x) = -\frac{1}{2} \log((2\pi e)^k |\Sigma_x|)$ from isotropic Gaussian residuals may lead to inconsistencies (see Figure 3.2). Figure 3.7 show the effect of different additions to the covariance (Equation (3.11)) in the pose estimation. We create a map from the very first RGB-D frame of sequence *fr2 xyz* [SEE⁺12a] and compute the available information to track each subsequent frame with respect to this initial map.

Figure 3.7 conveys at one glance the variation of the geometric covariance due to the propagation of depth uncertainty (Equation (3.10)) and due to perspective deformation (Equation (3.8)). Note that, for motions producing big parallax ($\cos(\alpha) \ll 1$) the depth covariance is dominant. On the other hand, approximations ($\frac{z_i}{z_0} < 1$) or distancing motion ($\frac{z_i}{z_0} > 1$) produce strong perspective deformations. Our covariance model bridges the gap between visual errors and meaningful entropy values of the state.

3.7.2 Photometric odometry

Photometric BA. ID-RGBDO [FCT20] is a RGB-D direct odometry that uses information metrics for informative point selection and keyframe creation. ID-RGBDO performs BA over cameras and points in a sliding window. We implement at the end of each run a photometric global BA over all keyframes and points used along the sequence. We run the BA iteratively over this seed modifying poses and points with small Gaussian noise to observe the error

Sequence	[FCT20]	ours	Sequence	[FCT20]	ours
fr1. xyz	1.55	1.62	fr3. tex. str. far	1.56	1.45
fr1. rpy	7.30	6.20	fr3. tex. str. near	1.89	1.78
fr2. xyz	0.90	0.81	fr3. tex. nstr. near	3.89	3.52
fr2. rpy	0.72	0.63	fr3. tex. str. far. v.	1.22	1.98
fr2. desk	2.15	1.88	fr3. tex. str. near. v.	4.50	2.36
fr2. dishes	7.07	5.02	fr3. tex. nstr. near v.	6.56	2.92
fr3. long office	3.11	2.65	fr3. long office v.	2.95	2.26

Table 3.4: ATE (cm) for photometric BA in different sequences of TUM RGB-D. For each pair, the left one is the baseline with isotropic noise and the right one with our deformation model. Ours outperforms the baseline in 12/14 sequences, with an average ATE reduction of 12.6% and a maximum reduction of 55.5% in *fr3. tex. nstr. near v.*

distribution. We evaluate two models for the residual covariance: an isotropic Gaussian one and ours, based on deformations. Table 3.4 collects the results in a selection of sequences of the TUM RGB-D dataset [SEE⁺12a]. Specifically, we use all static sequences where the accuracy of the resulting trajectory is enough to guarantee that photometric BA converges, taking into account the smaller baseline for direct methods to converge. Note that our deformation model consistently leads to smaller trajectory errors.

3.7.3 Feature-based SLAM

Feature-based BA. ORB-SLAM2 [MAT17a] is a feature-based SLAM system for monocular, stereo and RGB-D cameras. It includes some capabilities like map reuse, loop closing and relocalization. We run ORB-SLAM2 (where loop closure was deactivated from the original implementation in [MAT17a]) in different sequences and we apply a global BA at the end of each sequence over all the map points and all the keyframes poses. We modify this map by adding small Gaussian noise, in order to show variability in different runs. We then evaluate in different sequences two configurations for the global BA: with and without our deformation model. Table 3.5 shows the absolute trajectory error (ATE) of both configurations. Compared to the ATE of photometric BA (Table 3.4), notice two things. First, the tighter distribution of errors, confirming the better convergence of feature-based methods. And second, a smaller improvement, due to a higher degree of maturity of these methods and the complexity of modeling accurately the effect of the feature processing.

Sequence	[MAT17a]	ours	Sequence	[MAT17a]	ours
fr1. xyz	1.34	1.13	fr3. tex. str. far	1.08	0.90
fr1. rpy	3.17	3.09	fr3. tex. str. near	2.12	1.96
fr2. xyz	0.54	0.54	fr3. tex. nstr. near	1.37	1.21
fr2. rpy	0.37	0.35	fr3. tex. str. far. v.	1.12	1.04
fr2. desk	4.15	3.99	fr3. tex. str. near. v.	1.35	1.10
fr2. dishes	4.67	4.47	fr3. tex. nstr. near v.	1.56	1.51
fr3. long office	2.39	2.33	fr3. long office v.	2.34	2.06

Table 3.5: ATE (cm) for feature-based BA in different sequences of TUM RGB-D. For each pair, the left one is the baseline with isotropic noise and the right one with our deformation model. Ours outperforms the baseline in 13/14 sequences, with an average ATE reduction of 9.7% and a maximum reduction of 22.7% in *fr3. tex. str. near. v.*.

3.8 Conclusions and Future Work

In this chapter we have derived for the first time a general model for the perspective deformation of 2-dimensional image patches and, based on that, we have particularized the relation of this deformation with feature-based and photometric residuals. We have validated the goodness of fit of the model in both synthetic and real data, and we have shown experimentally that including perspective deformation into residual covariances improves the accuracy of direct and feature-based odometry and SLAM at a negligible computational cost and with minimal integration effort. Up to our knowledge, this is the first time that perspective deformation is explicitly modeled and applied to odometry and SLAM. We also show how to obtain more meaningful information metrics by modelling the covariances of the perspective deformation. Our evaluation focuses on global BA; since it is not coupled with other real-time parts of the pipelines (*e.g.*, keyframe creation) and hence removes other factors from the evaluation.

Chapter 4

■ SID-SLAM: Semi-Direct

Information-Driven RGB-D SLAM

4.1 Abstract

This chapter presents SID-SLAM, a complete SLAM framework for RGB-D cameras. Our main contribution is a semi-direct approach that, for the first time, combines tightly and indistinctly photometric and feature-based image measurements. Additionally, SID-SLAM uses information metrics to reduce the state size with a minimal impact in the accuracy. Our evaluation on several public datasets shows that we achieve state-of-the-art performance regarding accuracy, robustness and computational footprint in CPU real time. In order to facilitate research on semi-direct SLAM, we also contribute the Minimal Texture dataset, composed by RGB-D sequences that are challenging for current baselines and in which our pipeline excels.

4.2 Introduction

Visual odometry and SLAM systems are typically divided in the literature into two categories, *feature-based* and *direct methods*, depending on the type of residuals that are minimized [CCC⁺16]. But, **why should we choose between the two?** Our contribution in this chap-

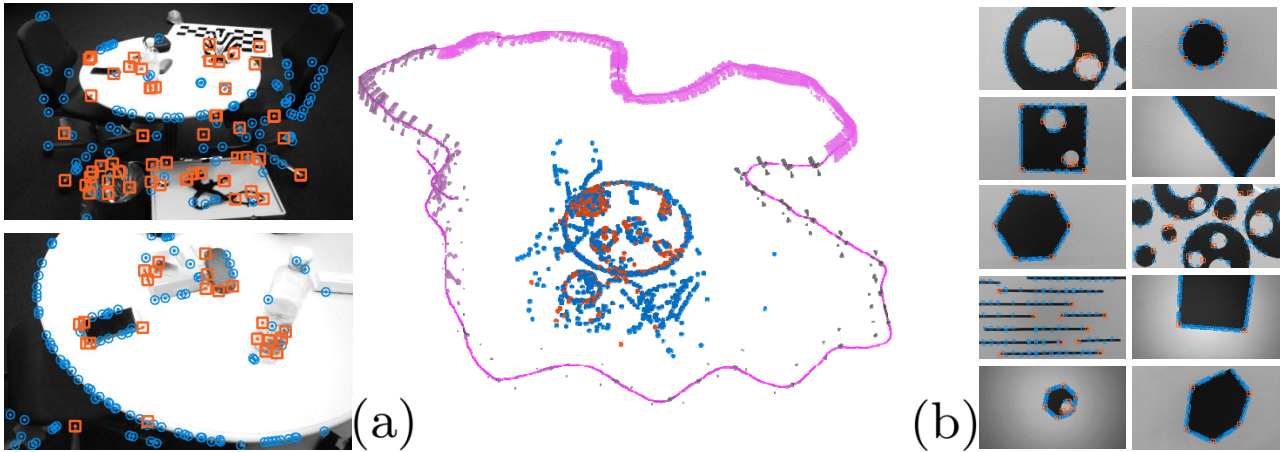


Figure 4.1: (a) Features (orange squares) and high-gradient pixels (blue circles) tracked by SID-SLAM and estimated map in a ETH3D sequence [SSP19]. Jointly minimizing photometric and feature-based residuals improves SLAM robustness and accuracy, specially in scenarios such as our minimal texture dataset (b).

ter is a strategy to use indistinctly feature-based or photometric residuals, depending only on their information content, and minimizing them jointly to estimate the SLAM state. See Fig. 4.1 for an illustration of our results. We implemented SID-SLAM full RGB-D SLAM pipeline to evaluate our proposal. Our results demonstrate that an information-based tight fusion of photometric and feature-based residuals achieves state-of-the-art performance in accuracy, robustness and computational footprint. Our fusion of residuals is particularly useful for minimal texture cases. In order to illustrate that, we contribute a novel dataset which is conceptually simple, but extremely challenging for current RGB-D SLAM baselines.

The key of our proposal is the complementary nature of feature-based and photometric methods. We will elaborate further on this in the rest of this section. *Features* (e.g., corners, blobs) can be robustly tracked up to a certain degree of illumination and viewpoint changes. However, they appear sparsely in images and hence do not exploit all available information. In contrast, *direct methods* [IA99] use potentially all available data, since they use the raw pixel intensities. But their high variance to illumination and perspective changes, most of the times not accounted by the residual models, makes them fragile in practical applications. Rolling shutter effects, sensor asynchronism and calibration errors [YWGC18]) are also more problematic for direct methods than feature-based ones.

Features require data association, for which correspondences are searched independently first, and robust estimation deals with spurious matches later. Since feature detection and matching

runs at real time, most detectors are optimized for speed rather than precision. Direct methods do not need prior data association, since this is implicitly given by the geometry. This allows to track pixels on weak corners and edges, in environments with little or high-frequency textures (e.g., sand [MSFV⁺21] or asphalt). However, as a drawback, their convergence is limited to the basin of attraction of image gradients. An important difference to be highlighted is the geometric dimension of both minimization errors. The error is 2-dimensional in the case of features, whereas the alignment of an edge is restricted to its normal direction [FZG⁺16]. Also related to it, the main challenge for a successful fusion is a proper model of the residual covariances, that we address in our work.

4.3 Related Work

The taxonomy of modern VO/SLAM into *feature-based*, *direct* and *semi-direct* (or *hybrid*) has been extensively addressed in previous works [YWGC18, LC18, SSP19]. Here we focus only on *semi-direct* strategies. *Semi-direct* methods exploit the complementarity of both feature-based and direct methods, and the challenge is doing it without compromising the efficiency, accuracy and robustness.

Combining corners and higher-level features. Combining geometric features has been extensively explored. As a few examples, [FPS14, FZG⁺16] use reprojection errors of corners and edgelets, [VLF04, PVA⁺17, ZLK18, GOMZN⁺19, CCGFO21, ZWK21] combine in different manners points and lines, and [Kae15, MKSC16, ASC20, ZKK21] use points and planes in the state.

Loose coupling between photometric and feature-based residuals. There are several works in the literature that use features and direct methods in SLAM, but always at different parts of the pipeline and in a loosely coupled manner. [FPS14, FZG⁺16] used photometric alignment for tracking and pixel triangulation, and feature-based joint optimization of structure and motion. Similarly, [LC18] combines photometric bundle adjustment of the local structure and motion [EKC17] and geometric bundle adjustment for larger optimization windows [MAT17a]. Early direct SLAM algorithms [SB12, KSC13a, ESC14] used nearest neighbour search over keyframes for the loop closure. [CC17, GWDC18] added to a direct VO thread a bag-of-words loop closure [GLT12] and the optimization of a co-visibility graph of keyframe poses. This is

similar in [SSP19], but in this case the map optimization is done by an alternating direct Bundle Adjustment. Although all these works benefit from both point types, their loose coupling limits their performance compared to using the same landmarks in tracking, mapping, and relocalization tasks [MAMT15, CER⁺21]. Up to the authors' knowledge, only the early [GBN08] uses together photometric and image reprojection errors for the case of pairwise camera motion.

Colored ICP. Minimizing 3-dimensional distances together with photometric errors has been used in many RGB-D odometry/SLAM works, e.g., [WJK⁺13, KSC13a, PZK17]. Differently from us, they use both errors *always* and do not select the most informative one. Their relative weight is tuned experimentally in most cases, which might cause problems in domain changes.

4.4 Semi-Direct Model Formulation

This section will cover the necessary background, notation and contributions of our semi-direct formulation, and the specifics of our SID-SLAM and Minimal Texture dataset will be detailed in Section 4.5.

Points. We represent 3D map points $\mathbf{p} \in \mathcal{P}\{\phi \cup \mathbf{f}\} \in \mathbb{R}^3$ according to their image representation, that is, $\phi \in \mathbb{R}^3$ if they are represented by image patches, or $\mathbf{f} \in \mathbb{R}^3$ if they are represented by feature descriptors. The image coordinates and inverse depth of $\mathbf{p} \in \mathbb{R}^3$ in reference frame j are denoted as $\mathbf{u}_j \in \Omega$ and $d \in \mathbb{R}$, where Ω is the image domain. For photometric patches we store a set of intensity values spread in a pattern \mathcal{N}_ϕ centered in \mathbf{u}_j [EKC17].

Keyframes. A keyframe j is defined by its RGB-D channels, its 6-DoF camera pose as a transformation matrix $\mathbf{T} \in \mathbf{SE}(3)$, two brightness parameters $\{a_j, b_j\}$ and a set of reference points to track. The Lie-algebras pose-increments $\widehat{\mathbf{x}}_{\mathfrak{se}(3)} \in \mathfrak{se}(3)$, with $\widehat{\cdot}_{\mathfrak{se}(3)}$ being the mapping operator from the vector to the matrix representation of the tangent space [Str12], are expressed as a vector $\mathbf{x} \in \mathbb{R}^6$. During the optimization, we update the transformations at step (k) using left matrix multiplication and the exponential map operator $\exp(\cdot)$, i.e.,

$$\mathbf{T}^{(k+1)} = \exp(\widehat{\mathbf{x}}_{\mathfrak{se}(3)}) \cdot \mathbf{T}^{(k)}. \quad (4.1)$$

The image points \mathbf{u}_i and \mathbf{u}_j are related by

$$\mathbf{u}_i = \Pi(\mathbf{R}\Pi^{-1}(\mathbf{u}_j, d_j) + \mathbf{t}), \quad (4.2)$$

where $\Pi(\mathbf{p})$ and $\Pi^{-1}(\mathbf{u}, d)$ are the projection and back-projection functions and $\mathbf{R} \in \mathbf{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$ are the relative rotation and translation between frames.

Residuals. The squared photometric residual $r_i^2|_\phi \in \mathbb{R}$ of a patch $\phi \in \mathcal{P}$ is the sum of the squared intensity differences between all pixels \mathbf{u}_j in the pattern \mathcal{N}_ϕ projected in frame i , with the corresponding pixel intensities in its reference keyframe j , combined with a logarithmically parametrized scalar factor e^{-a} and a photometric bias b [EKC17],

$$r_\phi^2|_i = \sum_{\mathbf{u}_j \in \mathcal{N}_\phi} \left(e^{-a_j} (I_j(\mathbf{u}_j) - b_j) - e^{-a_i} (I_i(\mathbf{u}_i) - b_i) \right)^2. \quad (4.3)$$

The reprojection residual $\mathbf{r}_i|_f \in \mathbb{R}^2$ of a feature $f \in \mathcal{P}$ in frame i is the geometric difference between the landmark projection \mathbf{u}_i and its associated observation, $\hat{\mathbf{u}}_i$

$$\mathbf{r}_f|_i = \hat{\mathbf{u}}_i - \mathbf{u}_i. \quad (4.4)$$

Residual Covariances. We use the model in [FMCT22] to properly model the multi-view covariances $\sigma_\phi^2 \in \mathbb{R}$ and $\sigma_f^2 \in \mathbb{R}^2$ of the residuals in equations (4.3) and (4.4). The photometric covariance

$$\sigma_\phi^2 = \sigma_I^2 + G^2 \sigma_\varphi^2(\boldsymbol{\eta}_g, \varepsilon^2, d) \quad (4.5)$$

is a function of the image noise $\sigma_I^2 \in \mathbb{R}$ and a geometric term σ_φ^2 , propagated with the photometric gradient G , which depends on the gradient direction $\boldsymbol{\eta}_g \in \mathbb{R}^2$, the perspective deformation $\varepsilon^2 \in \mathbb{R}$ and the inverse depth $d \in \mathbb{R}$ (see [FMCT22] for details). Similarly the covariance of a reprojection residual

$$\sigma_f^2 = \sigma_u^2 + \sigma_\varphi^2(\varepsilon^2, d) \quad (4.6)$$

depends on the associated noise of the feature descriptor $\sigma_u^2 \in \mathbb{R}^2$ and the propagated geometric noise $\sigma_\varphi^2 \in \mathbb{R}^2$.

4.4.1 Informative Point Selection

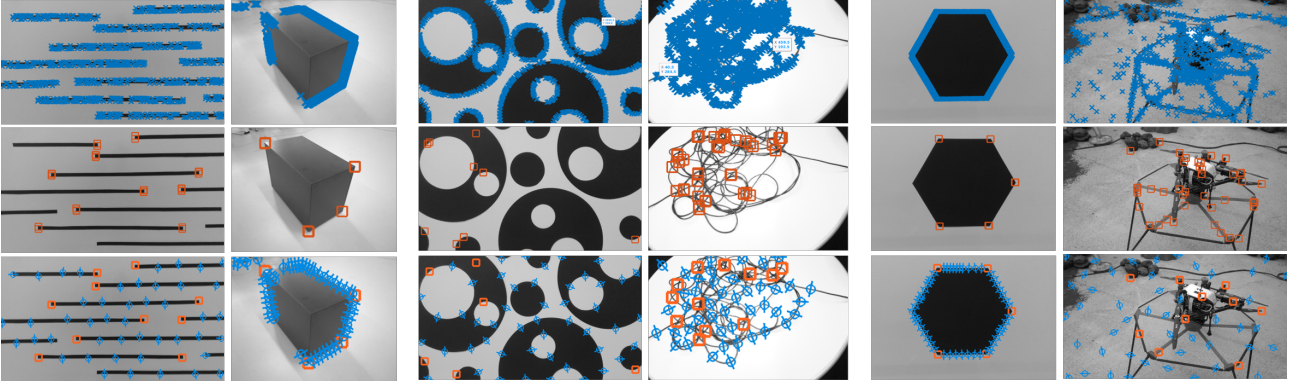


Figure 4.2: **Information-based point selection in RGB-D TUM [SEE⁺12a], ETH [SSP19] and our Minimal Texture dataset.** Top row: high-gradient points. Middle row: features. Bottom row: information-based selection.

We extend the approach in [FCT20] to select, in an iterative manner, the most informative points in an image. We analyze the contribution of each point \mathbf{p} to the accuracy of the camera pose \mathbf{x} in terms of entropy reduction [SMD10]:

$$E(\mathbf{x}) = \frac{1}{2} \log_2((2\pi e)^6 |\Sigma_x|), \quad \Delta_p E(\mathbf{x}) = \frac{1}{2} \log_2\left(1 + \frac{\Delta_p |\Lambda_x|}{|\Lambda_x|}\right), \quad (4.7)$$

where the information matrix

$$\Lambda_x = \Sigma_x^{-1} = \sum_{\phi \in Q} \mathbf{j}_\phi^T \sigma_\phi^{-2} \mathbf{j}_\phi + \sum_{f \in Q} \mathbf{j}_f^T \sigma_f^{-2} \mathbf{j}_f \quad (4.8)$$

is the inverse covariance matrix Σ_x^{-1} , obtained as the sum of the Jacobian auto-product for the whole set of selected points Q . ($\mathbf{j}_\phi \in \mathbb{R}^{1 \times 6}$) is the Jacobian of the photometric residual (4.3) with respect to \mathbf{x} . Analogously, ($\mathbf{j}_f \in \mathbb{R}^{2 \times 6}$) is the Jacobian of the features' residual (4.4). The variation to the information matrix determinant yielded by the addition of a photometric patch results in [FCT20]

$$\Delta_\phi |\Lambda_x| = \sigma_\phi^{-2} \mathbf{j}_\phi |\Lambda_x| \Lambda_x^{-1} \mathbf{j}_\phi^T, \quad (4.9)$$

that can be expressed individually per point, depending on \mathbf{j}_ϕ and the current inverse covariance matrix. From a first-order Taylor expansion of the determinant of the covariance matrix, we also estimate the contribution to the differential entropy for every feature

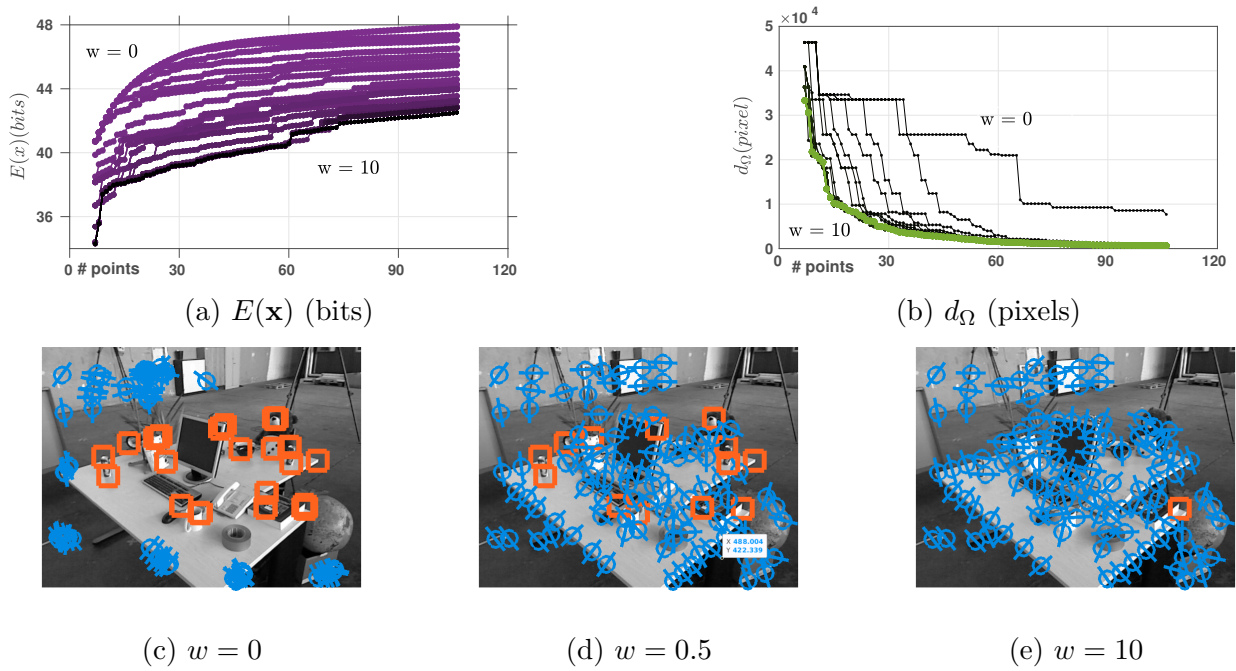


Figure 4.3: **Point Selection Strategy.** 4.3a and 4.3b show the variation, in consecutive steps of the algorithm, of the entropy of the camera pose $E(\mathbf{x})$ and the minimum image distance d_Ω between iteratively selected points for an interval of values of $w \in [0, 10]$. Note how increasing the influence of the image distance addend degrades the entropy contribution and vice-versa. Neglecting to spread points in the image (4.3c) concentrates the points in the most informative areas. Conversely, oversizing it (4.3e) neglects the informative content of each point. A trade-off between the two magnitudes balanced the selection strategy (4.3d).

$$\Delta_f |\Lambda_x| \approx |\Lambda_x| \Lambda_x^{-1} \cdot (\mathbf{j}_f^T \boldsymbol{\sigma}_f^{-2} \mathbf{j}_f). \quad (4.10)$$

We select iteratively points that maximize the trade off between their contribution to the camera pose entropy and their spreading in the image

$$s(\phi|\mathbf{f}) \Big|_k = \frac{\Delta_p E(\mathbf{x})}{\Delta_p E(\mathbf{x})|_{k=1}} + w \cdot \frac{d_\Omega}{\max d_\Omega|_k}, \quad (4.11)$$

where d_Ω is the distance for every point with respect to the closest already added point. Since entropy is a scene dependent metric, the first addend normalizes the contribution with the value obtained in the first iteration k . The second normalizes the image distances of the points with their maximum value on each iteration. We evaluated this selection method on a wide array of scenes (see Figure 4.2). Figure 4.3 shows the influence of the relative weight w in the point selection.

4.4.2 Information-based tracking

We track every frame reprojecting the points from a local map. We compute the normalized tracking information available per visible point from a reference keyframe with

$$\bar{E}(\mathbf{x}) = \log_2 \left| \left(\frac{\#\phi_r + \#f_r}{\#\phi_w + \#f_w} \right) \Lambda_x \right| = 6 \log_2 \left(\frac{\#\phi_r + \#f_r}{\#\phi_w + \#f_w} \right) + \log_2(|\Lambda_x|), \quad (4.12)$$

where $\#\phi_r$ and $\#f_r$ are the amount of visible points from the reference keyframe and $\#\phi_w$ and $\#f_w$ the total amount of visible points from the local map. Figure 4.4 shows how $\bar{E}(\mathbf{x})$ is used as a single threshold for keyframe insertion.

4.4.3 Bundle Adjustment with Semi-Direct Formulation

Semi-Direct joint residual. The full combined cost over all frames and points is given by

$$\sum_{j \in \mathcal{K}} \sum_{\phi \in \mathcal{P}_j} \sum_{i \in \text{obs}(\phi)} \left\| \alpha^2 (\sigma_i^{-2} r_i^2)_\phi \right\|_\gamma + \sum_{j \in \mathcal{K}} \sum_{f \in \mathcal{P}_j} \sum_{i \in \text{obs}(f)} \left\| \beta^2 (\mathbf{r}_i \sigma_i^{-2} \mathbf{r}_i)_f \right\|_\gamma, \quad (4.13)$$

where j iterates over all keyframes \mathcal{K} , ϕ and f over all points \mathcal{P} in keyframe j , and i over all frames $\text{obs}(\phi)$ and $\text{obs}(f)$ in which the point ϕ or f are visible. We apply a Cauchy robust cost function to decrease the influence of outliers scaled with a gamma probability value $\gamma_{0.95}$ [KSC13a][GOMGJ17].

Online covariance correction for residuals. Even if we use sophisticated uncertainty models (see equations (4.5) and (4.6)), non-modeled factors (such as motion blur or illumination changes) might unbalance the relative weight between photometric and reprojection residuals and impact the pipeline accuracy. We estimate at run time a correction factor for both residuals iteratively with the covariances α^2 , β^2 estimated online from the residual distribution

$$\alpha^2 = \gamma_\phi(\sigma_i^{-2} r_i^2 |_{j \in \mathcal{K}, \phi \in \mathcal{P}, i \in \text{obs}(\phi)}), \quad \beta^2 = \gamma_f(\mathbf{r}_i \sigma_i^{-2} \mathbf{r}_i |_{j \in \mathcal{K}, f \in \mathcal{P}, i \in \text{obs}(f)}), \quad (4.14)$$

where γ_ϕ and γ_f are the functions that map the covariance of gamma distributions from the median value of the residuals [KSC13a][GOMGJ17].

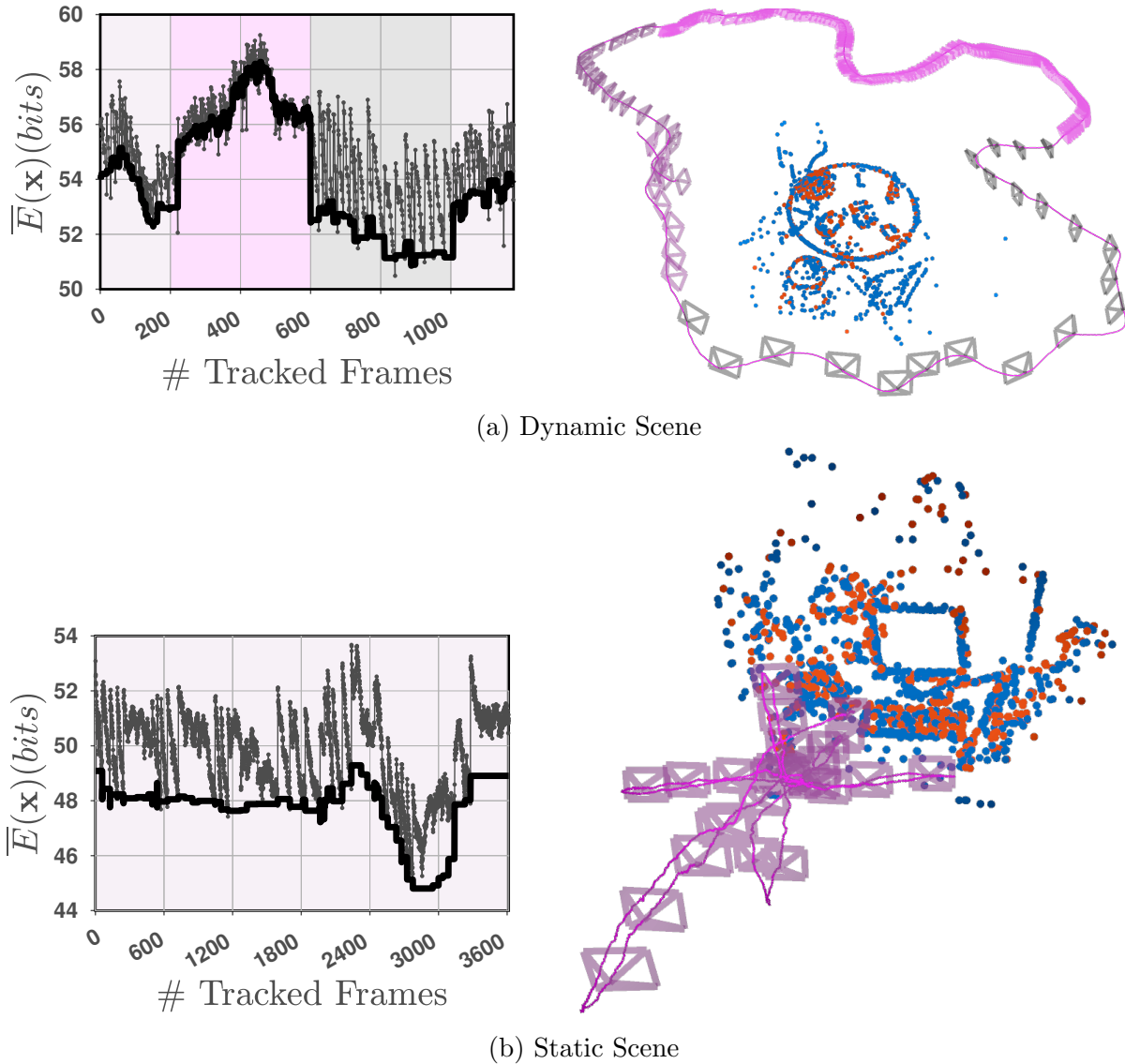


Figure 4.4: **Information Threshold for Keyframe Insertion.** **Left:** Exploratory trajectory. We manually modified four times the threshold to show how allowing bigger information drops reduces the keyframe creation speed. Darker cameras correspond to bigger information losses. **Right:** Non-exploratory trajectory. This sequence is longer than the previous one but, since the camera is not exploring new areas, our information criterion keeps a low number of keyframes. Note how the information reduces drastically as the camera moves away from the map.

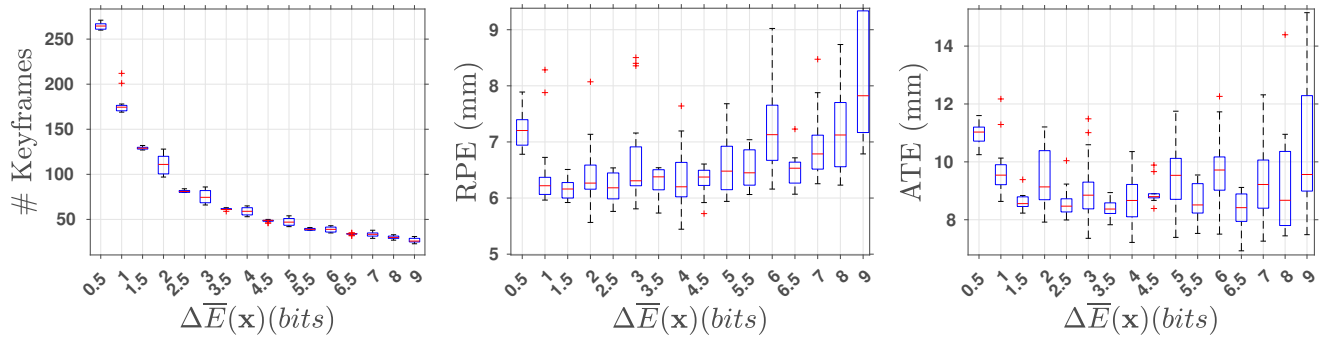


Figure 4.5: **Left:** Bigger drops of tracking information $\Delta \bar{E}(\mathbf{x})$ reduce the number of keyframe insertions. **Center:** Decreasing the number of keyframes deteriorates the relative pose error of the tracking. **Right:** The absolute trajectory error has a sweet spot with a 4-bit information drop. Bigger information drops reduce the tracking quality, and lower yield to trajectory drift.

Motion-only Optimization. As [EKC17], we jointly optimize the camera pose in SE(3) and brightness parameters with a coarse to fine pyramid resolution scheme.

Alternating Full BA. We use alternating optimization between cameras and points instead of jointly optimizing local and global full Bundle Adjustment. This facilitates the real-time performance of photometric Bundle Adjustment by speeding up Gauss-Newton optimization on strongly connected problems [PDL17, SSP19].

4.5 SID-SLAM

4.5.1 Windowed optimization

We track each new frame with respect to a reference keyframe and a local window of covisible keyframes around it. As detailed in Section 4.4.2, we insert a new keyframe when the tracking information drops more than a certain $\Delta \bar{E}(\mathbf{x})$ in bits.

We maintain a graph of covisible keyframes and, with every new keyframe, we trigger a Bundle Adjustment optimization of the cameras and points in a fixed-size window of the covisibility graph. For additional constraints, we project points of neighbouring keyframes in the optimization window.

Figure 4.5 shows the relationship between information loss and trajectory errors. The information loss threshold has a clear influence in the keyframes created. Increasing the frequency of

keyframe creation frequency improves the tracking quality (observe the RPE trend). However, an excessive number of keyframes reduces their geometric influence and the time available for local Bundle Adjustment. This increases the trajectory drift, and leads to a sweet tuning spot for minimizing the absolute trajectory error (observe the ATE graph).

4.5.2 Loop Closure, Pose Graph Optimization and Global BA

To correct the pose drift, we implemented a loop closure strategy that leverages both features and photometric intensities.

Loop detection is performed relying on the full set of AKAZE features extracted for each keyframe. Differently from classical bag-of-words approaches, that requires a carefully assembled vocabulary of features, we build upon HBST [SG18] where a binary tree of feature descriptors is built online and allows for efficient retrieval of similar images from a growing database. Following the insertion of keyframe i , the database is queried for keyframe j such that the number of occurrences of the same visual words is the highest. If the number of matches relative to the total number of features extracted is sufficient, we evaluate the number of co-occurrences with keyframes $j - 1$ and $j + 1$ looking for temporal consistency. As a first validation step of the candidate match, we match the full set of AKAZE features belonging to keyframes i and j to gather as many correspondences as possible. Then, a classical P3P-RANSAC step returns an initial transformation between the two keyframes.

Loop validation. Semi-Direct alignment is finally conducted as a last barrier against false positives and as a refinement of the estimated transformation, which is then utilized to bootstrap a global Bundle Adjustment step.

Loop closure. Performing alternating *semi-direct* BA instead of a full optimization might be still very costly, and therefore we perform it in three steps. (i) First a **pose graph optimization**. A pose graph optimization step, upon a loop detection, might push old keyframes out of the limited convergence region of the photometric part of the optimization. (ii) We have stored the geometric reprojection image position for all residuals, either photometric or feature-based, in all previous local windowed optimization. We incorporate to these measures the reprojections obtained in the loop validation step and perform **landmark-like pose BA**. (iii) Finally we perform **alternating full BA** adjustment to refine the global solution.

4.6 Experiments

Aleatoric effects and real-time constraints make performance comparisons between state-of-the-art SLAM pipelines challenging. In RGB-D SLAM, it is common practice to compare the Absolute Trajectory RMSE with SE(3) alignment (SE(3) ATE RMSE, [SEE⁺12a]). Among the good practices, (i) [SSP19] suggests evaluating in **complete benchmarks** instead of subselect (potentially cherry-pick) sequences, [MAT17a, CC17] compare **median results** (ii) over several runs to account the non-deterministic effects of multithreading, (iii) [DNZ⁺17] shows **memory consumption** (GB) for the captured sequences, and (iv) [FZG⁺16] shows the **processing time** (v) running the experiments in the **same machine**.

In this work we run our own evaluation of SID-SLAM, the feature-based baselines ORB-SLAM2 [MAT17a] and ORB-SLAM3 [CER⁺21], and the photometric baseline BAD-SLAM [SSP19] in three public RGB-D datasets (i). Tables 4.1, 4.2 and 4.3 gather the median value of the absolute trajectory error over ten runs per sequence (ii). We gather all values found in literature and also the running details. We compare memory footprint and resources consumption using a laptop with an Intel Core i7-10875H, 32 GB of RAM and an NVIDIA GeForce RTX 2070 8GB (iii),(iv),(v). We report the **percentage of trajectory** (vi) that baselines are able to compute. We only compare accuracy for runs that cover **rigorously 100%** (vii) of the sequence, ensuring that accuracy is compared also in challenging parts of the sequences. We collect the values of our evaluation together with those found in the literature and reflect relevant **evaluation conditions** (viii) about each evaluation. We also report the **number of keyframes** (ix) created per sequence which is intimately linked to accuracy and memory footprint. Finally we evaluate in our Minimal Texture dataset.

4.6.1 Results in public RGB-D datasets

Accuracy analysis. Figure 4.6 shows the percentage of trajectory estimated successfully by the three baselines and our SID-SLAM. Tables 4.1 and 4.2 report their accuracy in fully completed runs, avoiding misleading comparisons between runs that have been partially estimated. Values in bold represent the smallest tracking error per sequence, values in parentheses correspond to runs with at least 50% of the estimated track, and dashes represent large tracking errors early in the sequence.

RGB-D TUM dataset. Overall, dense approaches are more robust than sparse ones in extreme textureless sequences, as in *fr3 notex. near* or *fr3 large cabinet*. However, in scenes with small degrees of texture but with visible corners and edges, our semi-direct approach makes efficient use of all visual information and outperforms both dense and feature-based methods (as in *fr3 notex. far*). Similarly, in the *fr3 tex. far* sequences, where the scene content evolves from a high-frequency textured scene to a gradient-shaped cable, our approach outperforms all the baselines. Even in richly textured sequences, as *fr2 desk*, SID-SLAM outperforms the baselines. This is of high merit, as pure feature-based approaches avoid photometric noises and fusion nuisances and they should shine there. In summary, SID-SLAM improves robustness over feature-based methods by completing 24/31 sequences and achieves the best accuracy over all other baselines at 12/31 sequences.

ETH3D benchmark and Synthetic RGB-D TUM. BAD-SLAM [SSP19] consistently obtains the best accuracy in the ETH3D benchmark [SSP19]. This good performance is the result of two factors. Firstly, high-quality sensors calibrated with low errors downplay typical feature filtering that is so convenient with lower quality cameras. And secondly, BAD-SLAM’s additional minimization of a depth alignment residual. This helps in cases of poor visual information, which is beneficial in this high-quality dataset, but adds a dependency on the depth measurements that might introduce errors in lower-quality data. Our approach achieves similar performance on these sequences which are comparatively shorter (especially *plant* with less than 200 frames) and where the accuracy range is of the order of tenths of a millimetre.

ICL-NUIM. Our SID-SLAM complete all the sequences and obtains the best accuracy in 6/16 sequences in the living room and office environments. We believe the synthetic nature of the data, with non-informative planar depths in many cases, damaged BAD-SLAM performance.

Keyframe insertion and memory footprint. Figure 4.7 shows performance differences between the sparse feature-based, sparse semi-direct and dense photometric approaches. The bottom row of the heat map shows the number of inserted keyframes normalized to the number of frames per sequence. As can be observed in the table, our entropy-based criterion inserts the smallest ratio of keyframes, far from the BAD-SLAM ratio of more than 11 keyframes per frame. Note how ORB-SLAM2 and ORB-SLAM3 increase drastically the number of keyframes in some sequences to avoid tracking failure (as in *plant*).

Figure 4.7 reports the amount of memory allocated per keyframe for each baseline and our

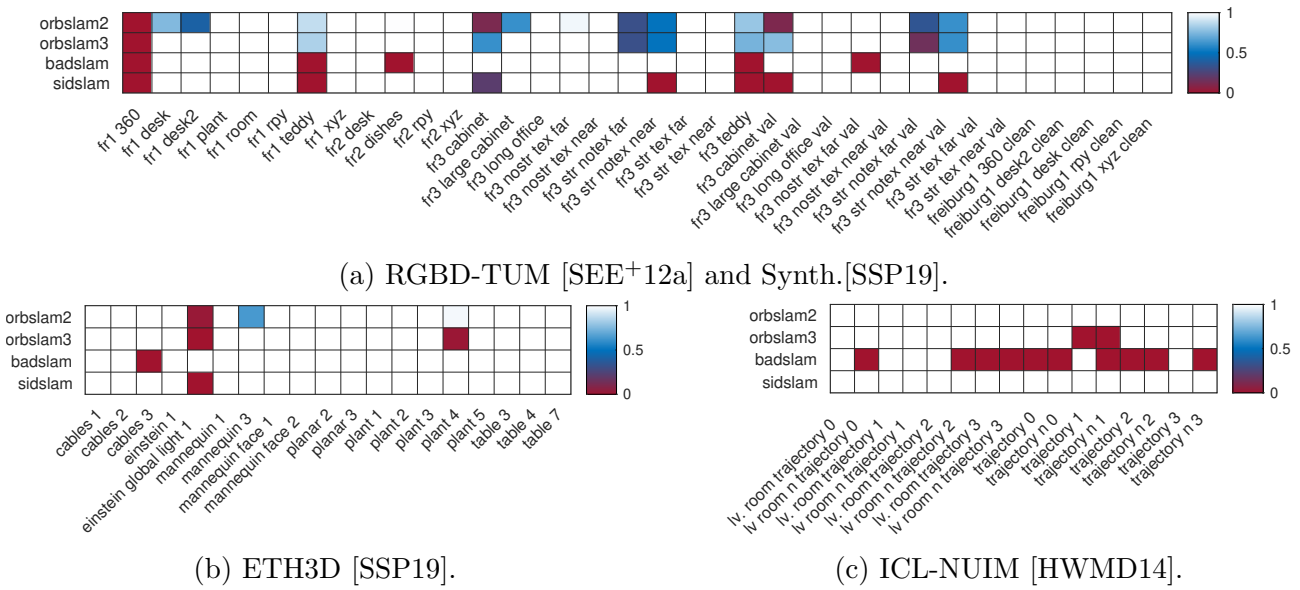


Figure 4.6: Percentage of estimated trajectory complete.

		360	desk	desk2	plant	room	tpy	teddy	xyz	desk	dishes	tpy	xyz	cabinet ¹⁴	large cabinet	long office	nostr. text. far	nostr. text. near	str. noext. far	str. noext. near	str. text. far	str. text. near	teddy	cabinet	large cabinet	long office	nostr. text. far	nostr. text. near	str. noext. far	str. noext. near	str. text. far	str. text. near			
		Freiburg 1													Freiburg 2				Freiburg 3				Freiburg 3 Validation ¹³												
[NIH ⁺ 11]	KinectFusion	\times^1	9.1	5.7					2.6																										
[EHE ⁺ 12]	RGB-D SLAM	\times^2	7.9	2.3	4.3	9.1	8.4	2.6	7.6	1.4	9.5 ¹¹	1.9	2.6 ¹¹				6.4			3.2 ¹⁰	1.7 ¹⁰														
[SB12]	MRSMap	\times	6.9	4.3	4.9	2.6	6.9	2.7	3.9	1.3	5.2	2.4	2.0							4.2 ¹⁰															
[EHS ⁺ 13]	RGB-D SLAM	-		2.6			8.7																												
[KSC13a]	DVO SLAM	\times	8.3	2.1	4.6	2.8	5.3 ¹²	2.0	3.4	1.1	1.7		1.8 ¹⁰				3.5		1.8 ¹⁰																
[NZIS13]	Voxel Hashing	\times^3		2.3							5.7	2.2					2.3		8.7																
[WKJ ⁺ 15]	Kintinuous	\times^4	3.7	7.1	4.7	7.5	2.8		1.7	3.4		2.9					3.0		3.1																
[WSMG ⁺ 16]	Elast. Fusion	\times^4	10.8	2.0	4.8	2.2	6.8	2.5	8.3	1.1	7.1	-	1.5	1.1	-	9.9	1.7	7.4	1.6	3.0	2.1	1.3	1.5	4.9											
[MAT17a]	ORB-SLAM2	\times^5		1.6	2.2	4.7					0.9		0.4 ⁸				1.0		1.9																
[CC17]	RGBD-TAM	\checkmark^6	10.1	2.7	4.2	2.5	15.5	2.1	8.1	1.0	2.7	3.6	0.2	0.7	5.7	7.0	2.7	2.6	1.0 ⁷	1.3	4.4	1.0	1.0	-											
[DNZ ⁺ 17]	Bundle Fusion	\times		1.6									1.1																						
[YYR17]	PSM SLAM	\times	5.5	1.6		5.1																													
[SSP19]	BAD-SLAM	\times		1.7 ⁹									1.1 ⁹																						
[MAT17a]	ORB-SLAM2	\checkmark	-	(1.6)	(2.2)	1.4	4.3	2.0	(3.9)	1.0	0.9	(4.3)	0.3	0.4	(6.3)	(4.9)	1.0	(5.9)	2.4	(0.9)	(2.5)	1.1	1.1	(1.4)	-	5.8	0.9	3.1	1.6	(1.2)	(2.0)	1.2	1.1	9/31	
[CER ⁺ 21]	ORB-SLAM3	\checkmark	-	1.8	2.7	2.0	7.6	2.2	(6.2)	1.0	1.8	8.8	0.5	0.4	(2.4)	15.1	1.1	4.6	2.1	(0.9)	(2.3)	1.0	1.0	(1.2)	-	5.4	1.0	3.5	1.5	(0.9)	(1.6)	1.2	1.1	8/31	
[SSP19]	BAD-SLAM	\checkmark	-	2.1	2.7	1.1	8.4	1.7	-	1.0	6.3	-	0.7	1.1	1.2	3.6	3.3	8.0	2.7	10.0	1.0	2.4	1.4	-	1.1	3.1	2.2	-	3.7	1.8	6.3	2.8	1.5	10/31	
	SID-SLAM	\checkmark	-	2.2	3.5	1.4	9.3	3.1	-	1.0	0.9	3.6	0.4	0.4	(6.3)	9.6	2.0	3.2	1.9	1.8	-	1.0	1.0	-	-	6.4	1.6	2.6	1.3	1.6	-	1.7	1.3	12/31	

Table 4.1: ATE (cm) in the TUM RGB-D benchmark [SEE⁺12a] for different baselines got from **their original publications** and from **our own evaluation**. We supply details about how the non-deterministic nature of the system is account (**nd.i**). **Notes on numbered entries:** (1) Evaluated in [KSC13a]. (2) Evaluated in [SB12]. (3) Evaluated in [DNZ⁺17]. (4) Best estimate over ten runs. (5) Using per-sequence best parameters. (6) Values are the median results over 5 runs of each sequence. (7) Typo in [CC17]. (8) Depth maps were compensated for a 4% bias. (9) This experiments were performed over the RGB and depth distorted images which might explain part of the degraded performance. (10) These values appear in [WKJ⁺15] but not in the original publication [KSC13a]. (11) These values differ in [WKJ⁺15]. (12) Typo in [MAT17a]. (13) These sequences were part of the dataset hidden validation but the ground truth is now available. (14) Depths and RGB images are misaligned in the *cabinet* sequence

		nd.i.	cables 1	cables 2	cables 3	einstein 1	einstein glc 1	mannequin 1	mannequin 3	mannequin face 1	mannequin face 2	planar 2	planar 3	plant 1	plant 2	plant 3	plant 4	plant 5	table 3	table 4	table 7
			ETH3D																		
[WSMG ⁺ 16]	Elast. Fusion	✓ ¹	1.18	1.51	-	2.83	1.11	9.41	-	-	1.0	1.06	3.47	0.80	0.82	1.77	1.17	1.07	-	1.25	-
[KSC13a]	DVO-SLAM	✓ ¹	0.44	-	-	0.51	0.86	3.60	2.05	0.52	0.34	0.24	9.76	0.84	0.28	0.97	0.29	0.49	0.82	1.82	0.69
[DNZ ⁺ 17]	Bundle Fusion	✓ ¹	2.24	9.51	-	2.89	1.62	-	-	1.34	0.91	0.34	0.35	-	0.41	-	-	-	1.73	-	1.02
[MAT17a]	ORB-SLAM2	✓ ¹	0.74	0.80	1.63	0.40	0.86	1.19	1.52	0.37	0.11	0.52	5.71	0.16	0.28	2.01	0.17	0.41	0.56	0.83	0.99
[SSP19]	BAD-SLAM	✓ ¹	0.68	0.51	6.19	0.30	0.29	-	0.55	0.39	0.10	0.30	0.33	0.19	0.14	0.18	0.14	0.17	0.24	0.22	0.28
[CER ⁺ 21]	ORB-SLAM3	✗ ^{1,2}	2.12	1.10	2.85	2.19	7.55	4.12	3.28	1.18	7.51	3.36	8.35	0.33	0.65	2.19	0.55	0.35	0.98	1.40	2.43
[MAT17a]	ORB-SLAM2	✓	0.77	0.98	2.42	0.48	-	1.29	^(2,8)	0.47	0.17	0.35	1.54	0.22	0.24	1.49	^(0,24)	0.40	0.52	0.79	0.92
[CER ⁺ 21]	ORB-SLAM3	✓	0.73	1.08	2.34	0.45	-	1.21	1.59	0.40	0.23	0.35	1.49	0.27	0.38	1.59	-	0.43	0.53	0.74	1.09
[SSP19]	BAD-SLAM	✓	0.61	0.63	-	0.32	0.69	0.44	0.44	0.41	0.13	0.26	0.33	0.21	0.17	0.21	0.20	0.20	0.26	0.26	0.29
SID-SLAM		✓	0.73	0.80	3.51	0.50	-	1.02	1.76	0.45	0.16	0.30	0.87	0.29	0.17	0.43	0.24	0.39	^(0.41) ³	^(0.56) ³	^(0.80) ³

		nd.i.	360	desk	desk2	tpy	xyz	long office	ns.t.n.	avg. ⁴	med. ⁴	
			Synthetic Freiburg (Clean)									
[MAT17a]	ORB-SLAM2	✓	0.50	0.31	0.31	0.11	0.14	1.31	2.16	0.69	0.47 ⁵	
[CER ⁺ 21]	ORB-SLAM3	✓	0.72	0.35	0.32	0.13	0.15	1.45	2.45	0.80	0.35	
[SSP19]	BAD-SLAM	✓	0.10	0.16	0.17	0.04	0.10	0.11	0.85	0.22	0.15 ⁵	
SID-SLAM		✓	0.49	0.34	0.18	0.11	0.17	1.5	0.90	0.53	0.34	

		nd.i.	lv0	lvn0	lv1	lvn1	lv2	lvn2	lv3	lvn3	traj0	trajn0	traj1	trajn1	traj2	trajn2	traj3	trajn3
			ICL NUIM															
[MAT17a]	ORB-SLAM2	✓	0.8	0.8	15.2	12.4	1.7	2.7	1.0	1.1	3.0	4.4	7.3	5.3	1.4	2.3	8.7	6.0
[CER ⁺ 21]	ORB-SLAM3	✓	7.6	8.5	19.4	17.1	1.7	5.1	2.3	1.7	3.6	10.3	-	-	1.4	2.2	7.3	2.3
[SSP19]	BAD-SLAM	✓	0.2	-	0.2	1.1	7.8	-	-	-	-	-	0.5	-	-	-	0.8	-
SID-SLAM		✓	0.8	0.7	22.3	6.7	2.1	2.4	1.2	1.0	2.8	3.7	9.8	6.8	1.6	1.9	9.5	4.5

Table 4.2: ATE (cm) in the ETH3D benchmark [SSP19], the synthetics RGB-D TUM dataset [SSP19] and ICL-NUIM [HWMD14]. We supply details about how the nondeterministic nature of the system is account (**nd.i**). **Notes on numbered entries:** (1) Results from the online leaderboard. (2) This evaluation of ORB-SLAM3 gets distorted results. (3) We get a consistent scale bias of 0.5%. We believe this is due to a misalignment between photometric patches and features on the image that propagates to the range of millimetres. (4) Average and median values of the seven sequences. (5) Values from [SSP19]. (6) This value is a typo in [SSP19].

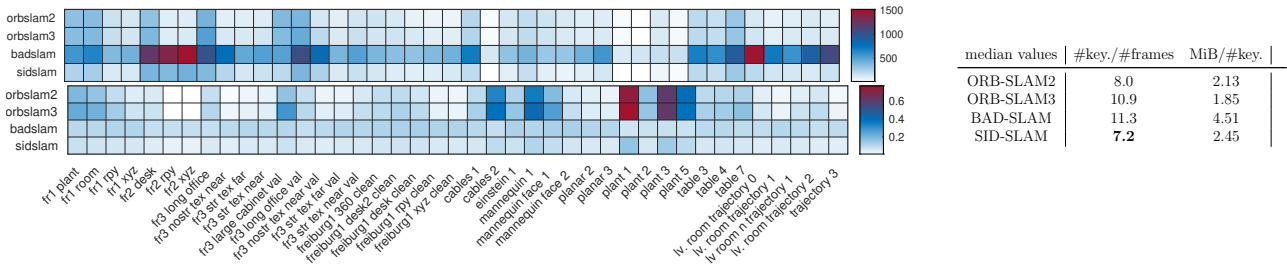


Figure 4.7: **Memory footprint.** **Bottom rows:** percentage of inserted keyframes in relation to the number of frames per sequence. **Top rows:** Total allocated memory in MiB per sequence. **Right Table:** median values of the keyframe percentage and the allocated memory per keyframe (Mib).

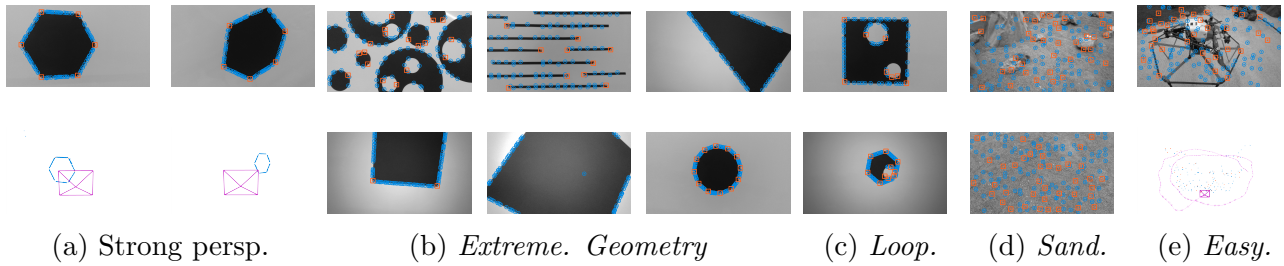


Figure 4.8: Representative frames from the **Minimal Texture dataset**.

SID-SLAM. BAD-SLAM is the most demanding in terms of memory and computation (note that it runs on GPU), as in addition to buffering gray and depth images it handles larger data loads. SID-SLAM uses slightly more memory per keyframe than the feature-based baselines (we need to buffer grey images but not their depth), but this is compensated by the lower keyframe ratio.

4.6.2 Results in Minimal Texture Dataset

Motivation. We recorded this new dataset to facilitate state-of-the-art research on semi-direct SLAM, particularly: (i) a better understanding of visual uncertainties of both features and photometric approaches [FMCT22], (ii) the efficient use of all the information on the image which maximizes SLAM robustness and reduces its computational footprint [FCT20] [APSL08].

Our dataset consists of 16 sequences with conceptually simple but challenging content. We group the sequences as *Extreme Geometry*, *Loop*, *Sand*, and *Easy*. The *Easy* set contains the control sequences to give an indicative measure of accuracy. *Extreme Geometry* sequences form the core of the dataset, focusing on minimal geometric content and strong perspective changes. The *Loop* set alternates between conceptual geometry content and the laboratory environment.

		Lines	Circle	Dodecagon	Hexagon	Square	Triangle 1	Triangle 2	Triangle 3	Circle Dodecagon	Circle Hexagon	Circle Square	Sand Rocks 1	Sand Rocks 2	Sand Rocks 3	Airlea	LRU	
		Ext.Geometry			Loop			Sand			Easy							
		14/11/21																
nd.i.																		
[MAT17a]	ORB-SLAM2	✓	1.5	(1.2)	(10.0)	-	-	-	-	(13.0)	(10.6)	(6.2)	6.6	-	3.5	1.8	5.5	
[CER ⁺ 21]	ORB-SLAM3	✓	1.5	1.1	(8.9)	1.9 ²	-	-	-	(13.7)	-	7.2	6.6	-	3.8	1.9	5.8	
[SSP19]	BAD-SLAM2	✓	-	-	-	-	-	-	-	-	-	-	8.0	8.5	3.8	10.0	-	
	SID-SLAM (ϕ) ¹	✓	-	0.8	2.7	3.0	(2.5)	1.9	3.1	2.4	6.6	7.1	4.7	5.8	7.6	3.9	2.1	6.3
	SID-SLAM (f) ¹	✓	2.1	-	-	-	-	-	-	-	-	-	-	-	5.1	4.5	7.2	
	SID-SLAM	✓	1.1	0.9	2.3	1.4	(2.3)	1.9	3.2	2.1	6.2	10.0	4.4	5.2	7.8	3.5	2.0	5.8

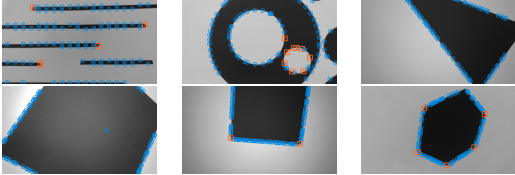


Table 4.3: ATE (cm) in **Minimal Texture** for different baselines and SID-SLAM. **Notes on numbered entries:** (1) These values are part of an ablation study and are therefore not suitable to compete with those of state-of-the-art baselines. (2) This is a modified version of ORB-SLAM in which we make it work with a minimum number of features.

Finally, the *Sand* group is meant to test *semi-direct* SLAM algorithms in textureless scenarios such as with planetary exploration purposes [MSFV⁺21]. The dataset was recorded with a Realsense D435i, capturing intensity and depth images of resolution 1920×1080 at rate of 30 Hz. We used a ceiling-mounted Vicon system to record millimeter-level ground truth for the camera pose.

Ablation study. We ablated SID-SLAM in two configurations: using only patches ϕ , and using only features f . The features-only configuration failed in all *Extreme geometry* sequences (*Triangles*, *Square*, *Hexagon* or *Dodecagon*) due to the low number of keypoints which, as can be seen in *Triangle* and *Square* images in Table 4.3, is occasionally reduced to none. Even SID-SLAM fails in *Square* as some configurations are quasi-degenerate. Finally, the patches-only configuration failed in *Lines* because, once again in the image, the photometric gradients were vertically aligned and it was only features placed at the extremes avoiding drift optimization on the horizontal axis. Note that the best accuracy in this sequence is obtained by complete SID-SLAM which grabs the necessary scattered features and refines the solution with photometric vertical gradients.

Evaluation. Table 4.3 shows that state-of-the-art baselines, both feature-based and photometric, fail at *Extreme geometry* sequences. This is caused by their inability to extract and process visual information. The reduction of thresholds for feature extraction and matching in ORBSLAM2/3 in sequences with just one *Square* (and thus only four corner-like features) leads to system failure. BAD-SLAM failed in all the geometry sequences. SID-SLAM outperforms all methods significantly both in robustness and accuracy.

4.7 Conclusions

In this work we present SID-SLAM, a complete RGB-D SLAM pipeline that, for the first time, fuses feature-based and direct methods in a tightly-coupled manner. As key contributions of our pipeline, we developed covariance models and information-based procedures for appropriate selection of the most informative points independently of its type and their fusion in a single cost function. We also use information criteria for keyframe selection. A thorough validation in three public datasets demonstrate that our SID-SLAM achieves state-of-the-art accuracy-robustness-efficiency performance. We further show the strengths of combining feature-based and direct methods in our novel Minimal Texture dataset, which also illustrates significant limitations in the literature.

Chapter 5

DOT: Dynamic Object Tracking for Visual SLAM

5.1 Abstract

In this chapter we present DOT (Dynamic Object Tracking), a front-end that added to existing SLAM systems can significantly improve their robustness and accuracy in highly dynamic environments. DOT combines instance segmentation and multi-view geometry to generate masks for dynamic objects in order to allow SLAM systems based on rigid scene models to avoid such image areas in their optimizations.

To determine which objects are actually moving, DOT segments first instances of potentially dynamic objects and then, with the estimated camera motion, tracks such objects by minimizing the photometric reprojection error. This short-term tracking improves the accuracy of the segmentation with respect to other approaches. In the end, only actually dynamic masks are generated. We have evaluated DOT with ORB-SLAM 2 [MAT17b] in three public datasets. Our results show that our approach improves significantly the accuracy and robustness of ORB-SLAM 2, especially in highly dynamic scenes.

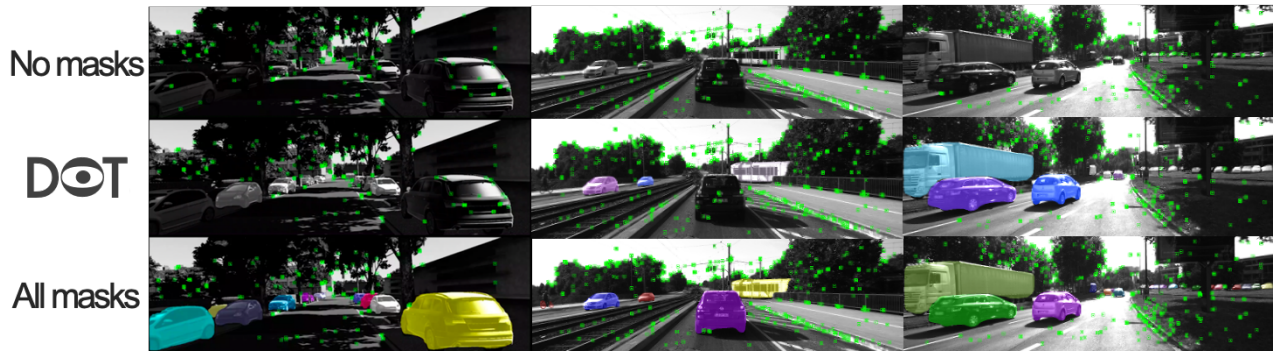


Figure 5.1: **Top row:** The frames correspond to ORB-SLAM2 [MAT17b] estimating the trajectory of the camera from the stream of images in The KITTI Benchmark [GLU12]. **Middle row:** Modified ORB-SLAM2 that works with the segmentation masks generated by DOT, which distinguish between moving and static objects. **Bottom row:** Modified ORB-SLAM2 using the segmentation masks provided by Detectron2 [WKM⁺19], that encode all potential dynamic objects. Note how from the most static scene (left column) to the most dynamic one (right column), DOT is capable to avoid moving objects while keeping the static ones. DOT achieves a trade-off between those two opposing scenarios by estimating the actual motion state of the objects in order to get higher tracking robustness and accuracy.

5.2 Introduction

Simultaneous Localization and Mapping, commonly known by its acronym SLAM, is one of the fundamental capabilities for the autonomous navigation of robotic platforms [CCC⁺16]. Its goal is the joint estimation of the robot motion and a map of its surroundings, from the information of its embedded sensors. Visual SLAM, for which the sensors are mainly, or exclusively, cameras, is one of the most challenging yet relevant configurations.

Despite the significant advances in SLAM in the last two decades, most state-of-the-art systems still assume a static environment, where the relative position between the scene points does not change and the only motion is done by the camera. With this assumption, SLAM models attribute the visual changes exclusively to the relative camera motion. A usual approach [MAMT15, MAT17b] is modeling dynamic areas as outliers, ignoring them during the pose tracking and map estimation processes. However, for several frames, until such dynamic areas are discarded as outliers, their data is used in the SLAM optimization, hence introducing errors and inconsistencies in the estimation of the map and the camera poses. Moreover, for feature-based SLAM methods, that track a small number of salient image points, the errors produced by a relatively small number of matches in dynamic areas are relevant and can lead to the system failure.

The world and the real applications in which a robot or an AR system must operate is far from being static. We can cite as representative examples the autonomous navigation of cars or drones, AR in crowded scenes or even planetary exploration tasks, where the poor texture makes SLAM systems precarious in the presence of shadows or other robots. Developing SLAM systems that are sufficiently robust to operate in highly dynamic environments is then essential for many applications.

As shown in the Figure 5.1, this work aims to develop an image processing strategy that improves the robustness of a visual SLAM system in dynamic environments. Our specific contribution is the development of “Dynamic Object Tracking” (DOT), a front-end that combines instance segmentation with multi-view geometry to track the camera motion, as well as the motion of the dynamic objects, using direct methods [EKC17]. The result of this pre-processing is a mask containing the dynamic parts of each image, that a SLAM system can use to avoid making correspondences in such regions.

Our experimental results in three different public datasets show that our combination of semantic segmentation and geometry-guided tracking outperforms the state of the art in dynamic scenes. We also find relevant that DOT is implemented as an independent front-end module, and hence easy-to-plug in existing SLAM systems. As DOT includes short-term mask tracking, we avoid the segmentation of all frames in the sequence, with significant savings in computation. Finally, although we tuned and evaluated DOT for the specific domain of car navigation, our strategy would be valid for other applications.

5.3 Related Work

SLAM in dynamic environments is an open research problem with a large scientific bibliography. We will divide the different approaches into three main categories.

The first category, and the most general one, models the scene as a set of non-rigid parts, hence including deformable and dynamic objects [NFS15, IZN⁺16, LPBM19]. While this research line is the most general, it is also the most challenging one. In this work we will assume intra-object rigidity, which is the premise behind the other two categories of dynamic visual SLAM.

The second category aims to improve the accuracy and robustness of visual SLAM by recon-

structuring only the static part of a scene. Dynamic objects are segmented out and ignored for camera pose tracking and map estimation. Along this line, DynaSLAM [BFCN18], built on top of ORB-SLAM2 [MAT17b], aims to estimate a map of the static part of the scene and re-use it in long-term applications. Dynamic objects are removed by combining 1) semantic segmentation for potentially moving objects, and 2) multi-view geometry for detecting inconsistencies in the rigid model. Mask R-CNN [HGDG17] is used for semantic segmentation, which detects and classifies the objects in the scene into different categories, some of which have been pre-set as potentially dynamic (*e.g.*, car or person). DynaSLAM was designed to mask out all the potentially mobile objects in the scene, which results in a lower accuracy than the original ORB-SLAM2 in scenes containing potentially mobile objects that are not actually moving (*e.g.*, scenes with many parked cars). The aim of this work is, precisely, to overcome this problem as only those objects that are moving at that precise moment will be labeled as dynamic.

Another work that has a similar approach is StaticFusion [SJP⁺], a dense RGB-D visual SLAM system where segmentation is performed by using the 3D reconstruction of the scene background as a way of propagating the temporal information about the static parts of the scene.

Finally, the third line of work in dynamic visual SLAM, which goes beyond the segmentation and suppression of dynamic objects, includes works such as MID-Fusion [XLT⁺18], MaskFusion [RBA18], DynSLAM [BLPG18] and ClusterVO [HYZ⁺19]. Their aim is to simultaneously estimate the poses of the camera and multiple dynamic objects. For that purpose, in MID-Fusion [XLT⁺18] and MaskFusion [RBA18] sub-maps of each possible moving object are created and a joint estimation of both the objects and camera poses is carried out.

Most of the systems mentioned [XLT⁺18, BLPG18, RBA18, HYZ⁺19, BFCN18] involve deep learning methods, which in some cases cannot be currently implemented in real-time due to bottleneck imposed by the limited frequencies of the segmentation network. The contribution developed in this work eliminates the requirement to segment all the frames, which allows the system to be independent of the segmentation frequency of the network, thus enabling its implementation in real time.

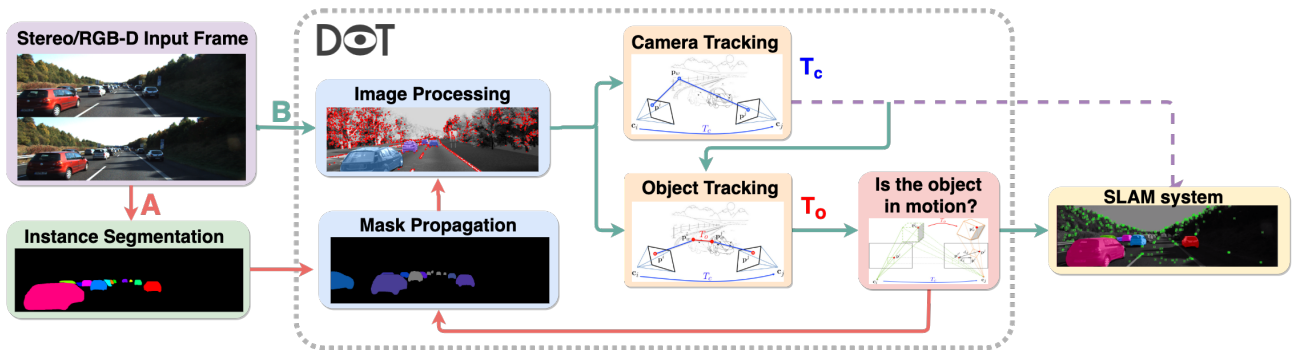


Figure 5.2: **Overview of DOT.** Path A (red), shows the processing for frames that get a segmentation mask from the network. Path B (green), shows the processing for frames that will acquire a segmentation mask geometrically propagated by DOT.

5.4 DOT

5.4.1 Overview

Figure 5.2 shows an overview of our proposal. The input to DOT are either RGB-D or stereo images at a certain video rate, and its output is a mask encoding the static and dynamic elements of the scene, which can be directly used by SLAM or odometry systems.

The first block (*Instance Segmentation*) corresponds to the CNN that segments out pixel-wise all the potentially dynamic objects. In our experiments, done using autonomous driving datasets, only cars were segmented as potentially moving. As it will be detailed later, since DOT tracks the mask from frame to frame, this operation does not need to be done at every frame.

The *Image processing* block extracts and separates the points belonging to static regions of the image and the points that are in dynamic objects. The camera pose is tracked using only the static part of the scene. From this block, and taking into account the camera pose, the motion of each of the segmented objects is estimated independently (*Object tracking*).

The next block (*Is the object in motion?*) determines, using geometric criteria, whether the objects labeled as potentially dynamic by the network are indeed moving. This information is used to update the masks encoding the static and dynamic regions of each frame and to feed the linked odometry/SLAM visual system.

Finally, DOT generates new masks from the estimations of the objects movement (*Mask Propagation*), so not every frame needs to be segmented by the network (see Figure 5.3). Given the

significant computational load of instance segmentation, this can be an relevant advantage of DOT compared to other state-of-the-art methods.

5.4.2 Instance Segmentation

We use the deep network Detectron2 [WKM⁺19] for the segmentation of all potentially movable instances that are present in an image. The output of the network has been modified to obtain in a single image all the segmentation masks. The image areas that are not classified into the potentially moving categories are given a ‘background’ label and are considered static in the subsequent blocks.

We use the COCO Instance Segmentation baseline model with Mask R-CNN R50-FPN 3x [LMB⁺14][Mat19]. The classes have been restricted to those considered as potentially movable, excluding humans since people tracking is beyond the scope of this work. In case other categories were needed, the network could be fine-tuned using these weights as a starting point or trained from scratch with its own dataset.

In order to consistently track the objects across multiple frames we have included a matching step between the masks computed by DOT and the ones provided by the net. New detections which cannot be paired with to any existing object are used to initialize new instances.

5.4.3 Camera and Object Tracking

From the instance segmentation of the previous step, we aim to estimate the motion of the camera and the dynamic objects. Since the motion of the camera and the motion of the objects are coupled in the images, we make the estimation in a two-step process. First we find the pose of the camera as a relative transformation $\mathbf{T}_c \in \mathbf{SE}(3)$ and then we subtract it to estimate the object motion $\mathbf{T}_o \in \mathbf{SE}(3)$.

Our optimization is related to the recent approaches of direct visual odometry and SLAM [EKC17], which aim to find the motion that minimizes a photometric reprojection error.

Optimization. Both for the calculation of camera pose and for the subsequent estimation of object motion, we do Gauss-Newton optimization

$$(\mathbf{J}^T \boldsymbol{\Sigma}_r^{-1} \mathbf{J}) \mathbf{x} = -\mathbf{J}^T \boldsymbol{\Sigma}_r^{-1} \mathbf{r}, \quad (5.1)$$

where $\mathbf{J} \in \mathbb{R}^{n \times 6}$ contains the derivatives of the residual function (equations (5.3) and (5.5)) and $\boldsymbol{\Sigma}_r \in \mathbb{R}^{n \times n}$ is a diagonal matrix containing the covariances of the photometric residuals $\mathbf{r} \in \mathbb{R}^n$. The Lie-algebras pose-increments $\widehat{\mathbf{x}}_{\mathfrak{se}(3)} \in \mathfrak{se}(3)$, with $\widehat{\cdot}_{\mathfrak{se}(3)}$ being the mapping operator from the vector to the matrix representation of the tangent space [Str12], are expressed as a vector $\mathbf{x} \in \mathbb{R}^6$. We update the transformations using left matrix multiplication and the exponential map operator $\exp(\cdot)$. Both optimizations are initialized with a constant velocity model and a multi-scale pyramid image to aid convergence.

Camera tracking. The camera motion is estimated using the static scene points \mathcal{P} and multi-view constraints [HZ03], assuming that the camera calibration and points depths are known. The projection of a static point $\mathbf{p} \in \mathcal{P}$ from its pixel coordinates \mathbf{p}^j in the reference frame F_j to its corresponding coordinates \mathbf{p}^i in the frame F_i is as follows:

$$\mathbf{p}^i = \Pi(\mathbf{T}_c \Pi^{-1}(\mathbf{p}^j, z_j)), \quad (5.2)$$

where Π and Π^{-1} correspond to perspective projection and back-projection models, respectively, and z_j is the depth of the point in the reference frame F_j .

The camera pose is optimized by minimizing the photometric reprojection error

$$\sum_{p \in \mathcal{P}} \left\| \left| I_j(\mathbf{p}^j) - I_i(\Pi(\exp(\widehat{\mathbf{x}}_{\mathfrak{se}(3)}) \mathbf{T}_c \Pi^{-1}(\mathbf{p}^j, z_j))) \right| \right\|_{\gamma}, \quad (5.3)$$

which is computed as the sum of all intensity differences between points in their reference frame and their projection into the frame being tracked. We use the Huber norm γ .

Object tracking. Once \mathbf{T}_c has been estimated, the pose of each potentially dynamic object can be estimated analogously by using the image points \mathcal{Q} belonging to such object. Modelling the potentially dynamic object as a solid with pose \mathbf{T}_o , the projection of each point $\tilde{\mathbf{p}}$ in the frame F_j to its coordinates in frame F_i is:

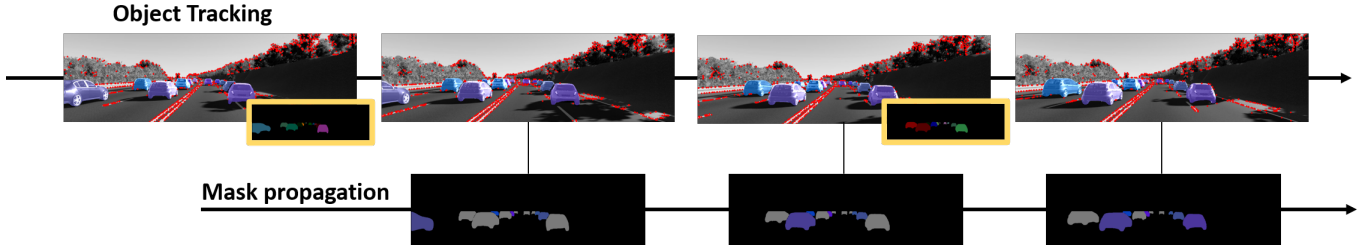


Figure 5.3: **Sample of a segment of the computation flow.** The upper row shows DOT estimating the tracking of the camera and the objects. Note how the segmentation masks from the network (yellow frames) are not necessary in all frames. The lower row shows the semantic masks generated by DOT that encode the motion classification: in motion (color), static (black) and not observed (gray).

$$\tilde{\mathbf{p}}^i = \Pi(\mathbf{T}_c \mathbf{T}_o \Pi^{-1}(\tilde{\mathbf{p}}^j, z_j)). \quad (5.4)$$

Analogously to equation 5.3, we estimate \mathbf{T}_o by minimizing the following photometric reprojection error

$$\sum_{\tilde{\mathbf{p}} \in \mathcal{Q}} \left\| I_j(\tilde{\mathbf{p}}^j) - I_i(\Pi(\mathbf{T}_c \exp(\hat{\mathbf{x}}_{sc(3)}) \mathbf{T}_o \Pi^{-1}(\tilde{\mathbf{p}}^j, z_j)) \right\|_{\gamma}. \quad (5.5)$$

5.4.4 Tracking quality, outliers and occlusions

Occlusions, changes in lighting conditions and segmentation errors have a significant effect in the accuracy of the objects and camera poses. As seen in algorithm 2, we developed several strategies that we apply after the object tracking step to reduce their impact.

Tracking quality. The appearance of dynamic objects changes significantly, producing high tracking errors. We used the Pearson’s correlation coefficient $\phi_o \in [-1, 1]$ to model appearance similarity. This metric reflects the degree of linear correlation between the reference intensities of the points and their corresponding estimates, hence being invariant to changes in gain and offset. Note that this metric can also be applied to camera tracking ϕ_c , although changes in the appearance of the background are usually less pronounced.

Outlier rejection. A common approach to detect outliers is defining an absolute threshold to the photometric error (5.3) (5.5). More sophisticated works [EKC17] adapt it according

Algorithm 2 Dynamic Object Tracking

```

1: function OBJECT TRACKING( $\mathcal{P}, \mathcal{Q}, \mathcal{O}$ )
2:                                      $\triangleright \mathcal{P}$  = static points
3:                                      $\triangleright \mathcal{Q}$  = dynamic points
4:                                      $\triangleright \mathcal{O}$  = set of objects
5:   mask  $\leftarrow \emptyset$                                       $\triangleright$  Dynamic mask to be computed
6:
7:    $\{T_c, \phi_c\} \leftarrow$  track camera ( $\mathcal{P}$ )                                      $\triangleright$  Camera Tracking
8:   if  $\phi_c < th_\phi$  then return  $\emptyset$ 
9:   end if
10:
11:  for object in  $\mathcal{O}$  do                                      $\triangleright$  Object Tracking
12:    if is visible (object,  $T_c$ ) then
13:       $\{T_o, \phi_o\} \leftarrow$  track object ( $T_c, \mathcal{Q}_o, \text{mask}$ )
14:      if  $\phi_o < th_\phi$  then break
15:      end if
16:      object  $\leftarrow$  outlier rejection ( $\phi_o$ )
17:      mask  $\leftarrow$  update mask (object)
18:      mask  $\leftarrow$  is object moving? (object)
19:    end if
20:  end for
21:
22:  return mask
23: end function

```

to the median residual, the motion blur or the lighting changes. As shown in Figure 5.4, we propose to set the threshold relative to the linear relation between intensities, so the errors are independent to photometric changes in the image.

Occlusions. The dynamic objects might occlude each other. Removing the occluded parts as outliers was not sufficient in our experiments. We implemented a strategy consisting of tracking the objects from the closest to the farthest, updating their respective masks sequentially. In this manner, we update in every iteration the points of the more distant objects that have been occluded by closer ones.

5.4.5 Is the object in motion?

This block receives as input the transformation matrices of the camera, \mathbf{T}_c and the objects, \mathbf{T}_o , and estimates whether the objects are moving or not. Its output, to be used by SLAM or odometry systems, are the masks that store the areas of the image occupied by dynamic objects

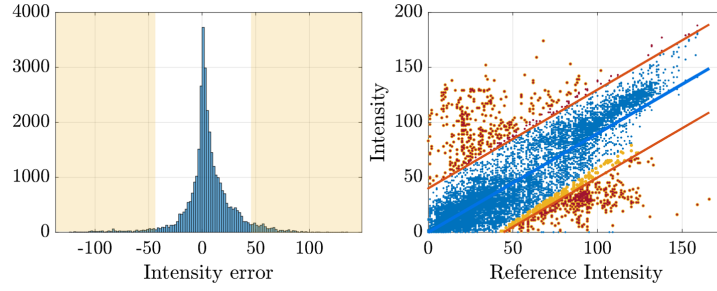


Figure 5.4: **Outlier rejection.** **Left:** histogram of photometric errors for an object. The shaded area corresponds to the points removed with a constant threshold. **Right:** Linear relation between intensities. Note the different points labeled as outliers by absolute (yellow) and relative (red) thresholds due to the changing photometry.



Figure 5.5: **Disparity vs Entropy.** Comparison of the dynamic disparities produced by different objects in motion. Note how observations with high entropy values (brighter red) produce larger shifts of image pixels.

and whether they are in motion or not. The masks are obtained by projecting the pixels of each object into the new frame using \mathbf{T}_c and \mathbf{T}_o estimated in the previous step.

Observing the object motion directly in \mathbf{T}_o generates, due to the propagated image noise, difficulties in establishing absolute thresholds that determine whether an object is in motion. In this work we chose to observe the motion of the objects using 2D image measurements. We denote our metric as *dynamic disparity*, being the distance in pixels between the projection of the point as if it were static \mathbf{p}^i and its actual projection $\tilde{\mathbf{p}}^i$. For each object we compute the median of the dynamic disparities of its points $\tilde{\mathbf{p}} \in \mathcal{Q}$:

$$d_d = \text{med}\{ \|\mathbf{p}^i, \tilde{\mathbf{p}}^i\|, \forall \tilde{\mathbf{p}} \in \mathcal{Q} \}. \quad (5.6)$$

The 3D motion of a point produces different image motions depending on 1) its image coordinates, 2) its depth, and 3) the relative angle between the directions of the object and the camera motions.

From the non-linear pose optimization (see eq. (5.1)) we can derive the uncertainty in the

estimation of the motion of the object $\Sigma_x = (\mathbf{J}^T \Sigma_r^{-1} \mathbf{J})^{-1}$. Assuming a k-dimensional Gaussian distribution, its differential entropy is:

$$H(\mathbf{x}_o) = \frac{1}{2} \log((2\pi e)^k |\Sigma_{x_o}|). \quad (5.7)$$

The differential entropy can be seen as the pose uncertainty derived from the photometric residuals minimization. In other words, observations of three-dimensional motions with high entropy values will result in larger shifts of image pixels (see Figure 5.5). On the other hand, observations with low entropy will produce small image disparities.

Based on this, the algorithm for classifying the movement of objects works as follows. We compare dynamic disparities (5.6) against a variable threshold $\Delta d = f(H(x))$ that grows smoothly with the entropy. We label as “in motion” all those objects whose dynamic disparity exceeds this threshold ($d_d > \Delta d$). For every value below an entropy threshold H_{min} we assume the object motion cannot be observed. Therefore, labeling an object as static requires that the motion is observable ($H(x) > H_{min}$) and that the median of the dynamic disparity is less than the variable threshold ($d_d < \Delta d$).

While selecting the optimal functional formulation would require further study, this expression meets the requirements and has shown good results in this work (see section 5.5.1). Figure 5.3 is an example of the mask propagated by DOT. Objects labeled as “in motion” are represented in colour, while those labeled as “static” disappear in black. The cars represented in gray are those which cannot be determined as being static neither dynamic.

5.4.6 Mask propagation

DOT exploits the two segmentation masks available in each frame: one produced by the neural network and other propagated from the previous frame. Warping one segmentation into the other allows to robustly relate instances found in different frames into the same 3D object.

State propagation. Relating new semantic instances to pre-existing objects allows us to predict their motion (which is critical for fast moving objects). In addition, it is possible to keep the classification of the motion in the case of an object moving to a position where the

motion is not observable (see Section 5.4.3).

Independent segmentation. Our proposal allows the propagation of semantic segmentation masks from an initial seed over time and space, eliminating the need for segmenting every frame. Running the neural network at a lower frequency makes real-time object tracking easier in low-end platforms. As further benefit, DOT is able to fill in the gaps in which the network temporarily loses the instantiation of an object between consecutive images.

5.5 Experimental Results

Although the potential applications of DOT cover a wide spectrum ranging from object detection to augmented reality or autonomous driving, in this chapter we provide an intensive evaluation to demonstrate to what extent “knowing the movement of objects” can improve the accuracy of a SLAM system.

5.5.1 Evaluation against baselines

Baselines. Our experiments estimate the camera trajectory using a state-of-the-art SLAM system in three different configurations. Specifically, we use ORB-SLAM2 [MAT17b], with its RGB-D and stereo implementations. The three configurations designed to evaluate DOT are:

No masks: ORB-SLAM2 is run using the authors’ implementation on unmodified images. A rigid scene is assumed, so all the points in the images (including those belonging to moving objects) can be selected by ORB-SLAM2.

DOT masks: ORB-SLAM2 receives as input, in addition to the images, the dynamic object masks containing potentially dynamic objects currently in motion. We modified ORB-SLAM2 so that it does not extract points from such moving objects.

All masks: ORB-SLAM2 receives all the masks obtained by the instance segmentation network. In this configuration, all potentially dynamic objects are removed without checking if they are actually moving or not.

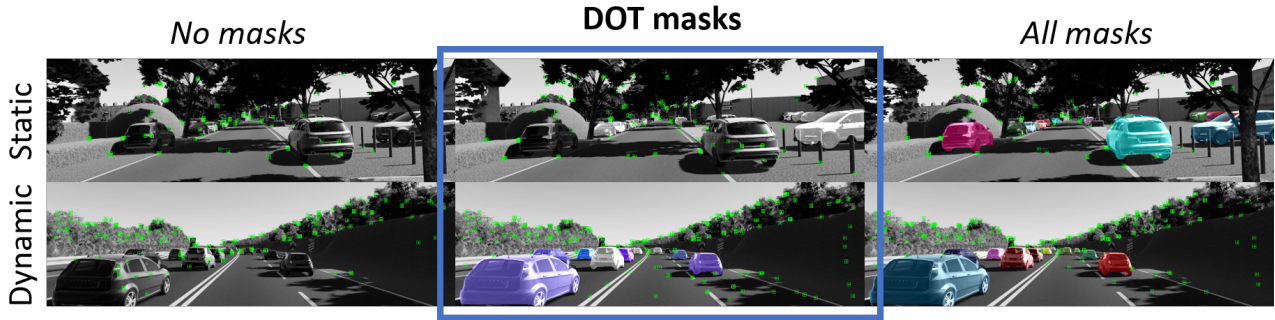


Figure 5.6: **Scene content adaptation.** Sample results for the three studied configurations. **Left:** *No masks*. **Centre:** *DOT masks*. **Right:** *All masks*. The top row shows a static scene in which the *All masks* setting discards all points of the static objects that can aid to tracking accuracy. In contrast, the bottom row shows how the *No masks* configuration allows to extract points on moving objects that may cause the system to fail. Both are cases in which the lack of understanding of the scene deteriorate the performance of SLAM. DOT successfully identifies the parked cars as static and the moving ones as dynamic. Note how DOT achieves a trade-off between those two opposing scenarios by estimating the actual motion state of the objects that results in a better estimation of the trajectory.

Seq.	ATE [m]			ATE/ATE _{best}		
	No masks	DOT	All masks	No masks	DOT	All masks
01	1.10	1.14	1.38	1.00	1.04	1.26
02	0.16	0.14	0.10	1.60	1.43	1.00
06	0.11	0.07	0.08	1.67	1.00	1.18
18	4.77	1.00	1.50	4.79	1.00	1.51
20	29.42	9.12	13.54	3.23	1.00	1.49
$\bar{\epsilon}_{norm}$	192.6%	100.0%	137.8%			

Table 5.1: DOT against baselines (*No masks* and *All masks*) in V-KITTI. **Left:** ATE [m]. **Right:** ATE over best ATE per sequence.

Sequence subsets. We evaluate the above configurations in three sequence subsets from the KITTI Vision Benchmark Suite [GLU12], containing stereo sequences of urban and road scenes recorded from a car and used for research in autonomous driving. We use Virtual KITTI [GWCV16] [CMH20], a synthetic dataset composed of 5 sequences virtually cloned from KITTI [GLU12], KITTI *Odometry*, a predefined subset of sequences specially designed for the development and evaluation of visual odometry systems, and a selection of sequences chosen from the *raw* section of KITTI because of their high number of moving objects [HYMH20].

We run the RGB-D version of ORB-SLAM2 in Virtual KITTI, as synthetic depth images are provided, while for the other subsets we run the stereo version of ORB-SLAM2 over the color stereo pairs. The ground truth for the real sequences is given by an accurate GPS localization

Seq.	ATE [m]			ATE/ATE _{best}		
	No masks	DOT	All masks	No masks	DOT	All masks
0	1.77	1.80	2.08	1.00	1.02	1.18
1	6.37	7.71	8.45	1.00	1.21	1.33
2	3.72	3.70	3.84	1.01	1.00	1.04
3	0.40	0.40	0.40	1.00	1.01	1.00
4	0.27	0.26	0.24	1.12	1.09	1.00
5	0.40	0.39	0.45	1.03	1.00	1.14
6	0.63	0.68	0.67	1.00	1.08	1.07
7	0.52	0.51	0.51	1.01	1.00	1.00
8	3.04	3.24	3.78	1.00	1.07	1.24
9	2.65	0.98	3.80	2.71	1.00	3.89
10	1.23	1.29	1.26	1.00	1.05	1.02
$\bar{\epsilon}_{norm}$	112.7%	100.0%	130.3%			

Table 5.2: DOT against baselines (*No masks* and *All masks*) in KITTI *Odometry*. **Left:** ATE [m]. **Right:** ATE over best ATE per sequence.

Seq.	ATE [m]			ATE/ATE _{best}		
	No masks	DOT	All masks	No masks	DOT	All masks
0926-0009	1.23	1.24	1.44	1.00	1.01	1.17
0926-0013	0.26	0.26	0.27	1.00	1.00	1.03
0926-0014	0.86	0.82	0.78	1.11	1.06	1.00
0926-0051	0.37	0.36	0.37	1.02	1.00	1.02
0926-0101	8.66	10.26	12.37	1.00	1.18	1.43
0929-0004	0.32	0.30	0.30	1.08	1.03	1.00
1003-0047	13.81	1.25	2.23	11.01	1.00	1.78
$\bar{\epsilon}_{norm}$	242.3%	100.0%	115.9%			

Table 5.3: DOT against baselines (*No masks* and *All masks*) in KITTI *Raw*. **Left:** ATE [m]. **Right:** ATE over best ATE per sequence.

system.

Evaluation metrics. As it is standard when evaluating real-time SLAM, in order to take into account non-deterministic effects, we run each configuration 10 times per sequence and report median values. All the experiments were run in a laptop with an Intel Core i5 processor and 8GB of RAM memory.

We report the absolute trajectory error (ATE) as proposed in [SEE⁺12b], which is the root-mean square error (RMSE) of the estimated position of all frames with respect to the GPS ground truth after both trajectories have been aligned. For an easier comparison between DOT and the other two configurations, we report the average of the errors normalized by the value

we obtained with DOT on each sequence $\bar{\varepsilon}_{norm} = \frac{1}{n} \sum_{i=0}^n \frac{\varepsilon_i}{\varepsilon_{DOT}}$.

The right columns in Tables 5.1, 5.2, 5.3 show the ATE normalized by the best ATE in each sequence among the three configurations. Thus, a value equal to 1 identifies the best result, while values > 1 are indicative of poorer performance. The color scale indicates the trade off of the errors between the best result (green) and the worst (red).

Tracking accuracy. The ATE in Table 5.1, corresponding to the V-KITTI sequences, show an accuracy improvement of 92.6% and 37.8% of our system with respect to the *No masks* and *All masks* configurations, respectively. In addition, DOT scores best for 3 of the 5 sequences evaluated.

Table 5.2 contains the ATE results for 11 trajectories of KITTI *Odometry* evaluated with the three different configurations. DOT obtains in this case an overall performance which is 12.7% and 30.3% better than *No masks* and *All masks*, respectively. Compared to V-KITTI, this group of sequences contains less dynamic elements, so the use of masks is even detrimental. According to the dataset specifications, the ground truth camera poses collected by the GPS are accurate to within 10 cm. Therefore, no significant differences exist between the three configurations in sequences 3, 4, 5, 6, 7 and 10. This is thought to be a consequence of the small number of moving objects, as well as of the rich texture of the images, which provides a large number of static points for estimating the camera motion.

The differences between sequences and methods are more evident with the last set of sequences shown in Table 5.3, characterized by an abundance of moving objects. Overall, DOT achieves improvements of 142.3% in ATE accuracy over *No masks* and 15.9 % over the *All masks* method. Again, note how discarding dynamic objects in sequence 1003-0047 reduces significantly the tracking errors. The sequences 0926-0009, 0929-0004 and 1003-0047 were cloned to generate the V-KITTI synthetic sequences (1, 18 and 20). As expected, since the scenes contents are identical, so is the qualitative analysis of the results.

The color scale used in Tables 5.1, 5.2, 5.3 shows how DOT tends to approach to the best solution when it is not the most accurate trajectory (green). This proves that, while the use of masks may be convenient, the accuracy is significantly improved if only the objects that have been verified to be in motion are removed. These results demonstrate that DOT achieves consistently a good performance both for static and dynamic scenes.

Scene content adaptation. Figure 5.6 illustrates two scenarios that affect the SLAM accuracy in a scene with dynamic objects. The lower row shows a road where all the vehicles are in motion (Seq. 20 in Table 5.1). The high dynamism of all the vehicles in the scene violates the rigidity assumption of ORB-SLAM2, and makes the system fail. Similarly, moving objects in sequence 18 (Table 5.1) causes tracking failure of ORB-SLAM2 in 6 out of 10 trials (only 56% of the trajectory could be estimated in those cases).

The upper row shows an urban scene with several cars parked on both sides of the road (Seq. 01 in Table 5.1). Contrary to the previous case, the worst configuration is using all the segmentation masks since a large number of points with high information content are removed for tracking. ATE results in Table 5.1 for this sequence shows that extracting points from a larger area results in a better accuracy of the estimated trajectory.

Summing up, notice how not using dynamic object masks increases the trajectory error due to matching points on moving objects. However, applying masks without verifying if the object is in motion discards a high amount of information, especially when a large part of the scene is occupied by vehicles. DOT achieves a trade-off between those two opposing scenarios by estimating the actual motion state of the objects in order to get higher tracking robustness and accuracy.

Loop closure. Not all differences in trajectory accuracy are due to poor tracking performance. The loop closure module of ORB-SLAM2 reduces the drift and therefore also the inaccuracies produced by dynamic objects or by the removal of parked vehicles. We have observed that ORB-SLAM2 running with *DOT masks* is able to close the loop 6 out of 10 runs in sequence 9 of KITTI *Odometry* (see Table 5.2), while none was identified when using *All masks*. This results in a broader error variability.

Segmentation errors. Compared to other approaches, DOT is capable of alleviating segmentation errors. Neural networks sometimes mislabel static objects (*e.g.*, traffic signs or buildings) as dynamic, DOT corrects this error by re-tagging the object as static (see Figure 5.7). As another example, when the network does not fire in one of the sequence frames, DOT is able to fill the gap by propagating the object mask.

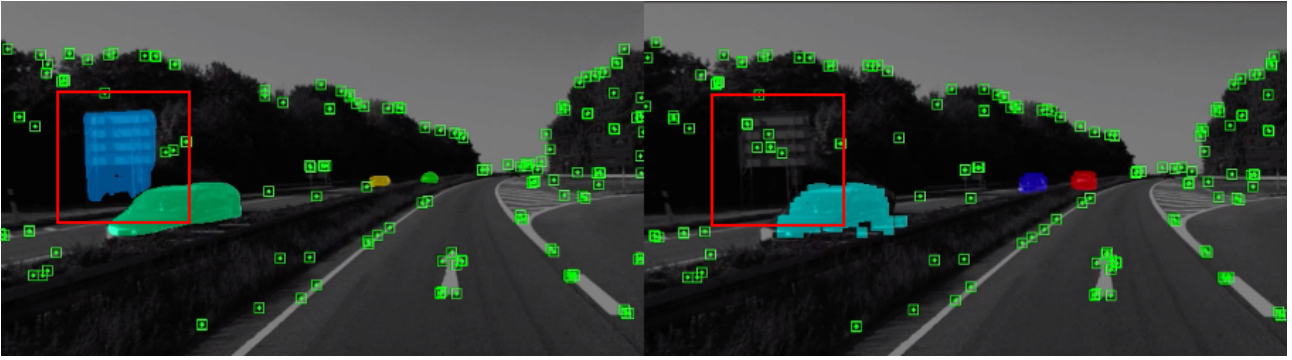


Figure 5.7: **Segmentation error.** Comparison between *All masks* and *DOT masks*. Notice that a wrong segment from Detectron2 (the sign in the red square is assigned a car label) is correctly classified as static by DOT.

5.5.2 Mask propagation

As explained in section 5.4.6, our approach allows to reduce the frequency of network segmentation by propagating pre-existing masks in the intermediate frames. Figure 5.8 shows the number of correctly labeled pixels minus mislabeled ones (ground truth in black) on every frame of V-KITTI when DOT uses 100% of Detectron2 segmentations (red), 50% (blue), 33% (yellow) and 25% (green). Note how the masks stay accurate when being propagated except when tracking failures occur or a moving object enters the scene between segmentations (see also intersection over union on V-KITTI in Table 5.4). We believe this result may be helpful for real time object tracking specially for high frequency image streams.

Rate	Seq01	Seq02	Seq06	Seq18	Seq20
1.0	0.88	0.88	0.84	0.90	0.89
0.5	0.74	0.83	0.67	0.85	0.84
0.33	0.72	0.80	0.60	0.85	0.81
0.25	0.69	0.78	0.55	0.84	0.81

Table 5.4: **Intersection over union** in the V-KITTI dataset for different segmentation rates.

5.6 Conclusions

DOT is a novel front-end algorithm for SLAM systems that robustly detects and tracks moving objects by combining instance segmentation and multi-view geometry equations. Our evaluation with ORB-SLAM2 in three public datasets for autonomous driving research

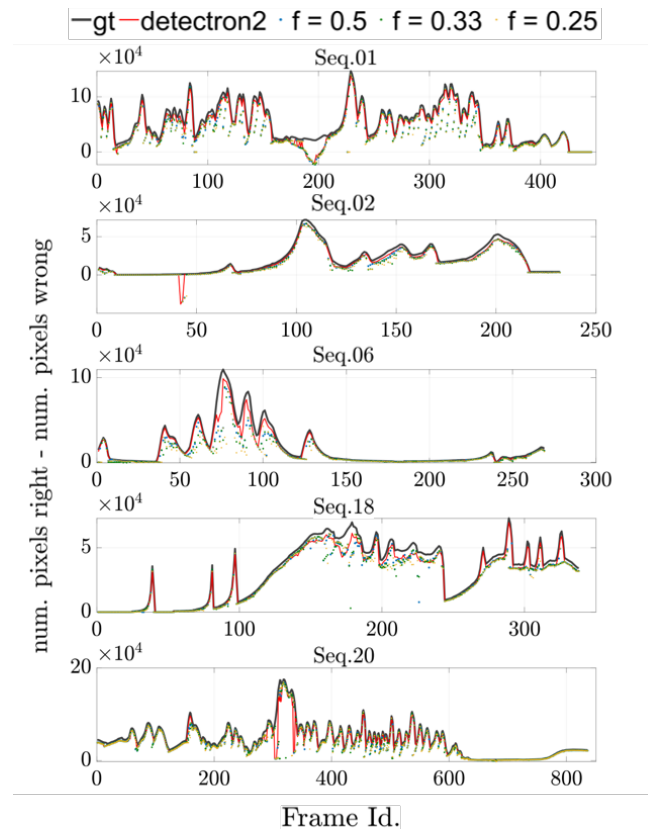


Figure 5.8: **Mask propagation.** We show for each frame of V-KITTI the number of correctly labeled pixels minus mislabeled ones respect to the ground truth (black), when DOT uses all masks from Detectron2 (red), 50% (blue), 33% (yellow) and 25% (green).

[GLU12][GWCV16][CMH20] demonstrates that DOT-generated object motion information allows us to segment the dynamic content, significantly improving its robustness and accuracy.

The independence of DOT from SLAM makes it a versatile front-end that can be adapted with minimal integration work to any state-of-art visual odometry or SLAM system. Unlike other systems, the mask tracking of DOT reduces the rate at which segmentation (typically involving high computational cost) should be done, reducing the computational needs with respect to the state of the art.

Chapter 6

Conclusion

6.1 Summary of Thesis Achievements

In this thesis, **Information-Driven Navigation**, we developed new probabilistic methods to improve accuracy, robustness and efficiency of Visual SLAM. The core of our work and contributions is issued in three main articles contained in chapters 2, 3 and 4. Along these publications we have thoroughly analyzed previous approaches and we have presented a significant number of new techniques which have been ablated and compared with the state of the art. The detailed descriptions of our state-of-the-art solutions, our open-source **SID-SLAM**¹ and the new **Mininal Texture dataset**¹ provide a solid base for future developments in the field.

As a summary of our work, for our first publication [FCT20] we applied information metrics in a visual odometry RGB-D framework based on direct methods with the aim of reducing its computational footprint. From the experiments and results performed in this work, we found that a better uncertainty model for visual measurements would lead to more realistic information metrics and became an instrumental element to our information-driven approach. In our second publication [FMCT22] we derived a covariance model for multi-view residuals from which the key element is the formulation of a term based on perspective deformation. Finally, we combined both contributions in the formulation of the first complete semi-direct RGB-D SLAM system, that uses tightly and indistinctly features and direct methods. We

¹We will publicly release the code and data after the ECCV review process to respect its double blind policy.

elaborate on these three contributions in more detail below along with our new Minimal Texture dataset and the publication of Dynamic Object Tracking (DOT) [BFC⁺21].

As mentioned above an important element of our visual SLAM solutions is an **information-theoretic approach to point selection**. In chapter 2 we have proposed a novel criterion to select the most informative points to be tracked in a RGB-D odometry framework. We have shown experimentally that using a small number of very informative points and keyframes can have a significant impact in the computational cost of RGB-D odometry, while keeping an accuracy similar to the state of the art. Specifically, our experimental results show that tracking the 24 most informative points is enough to match the performance of the state of the art while reducing the computational cost up to a factor $10\times$. Up to our knowledge, this is the first time that Information Theory is applied to direct odometry and SLAM methods. We believe that our results will facilitate the use of visual odometry and SLAM in small robotic platforms and AR/VR glasses, that are limited in computation and power.

A better model for the residual covariances will improve the accuracy of multi-view structure and motion estimations. Moreover, it will lead to more realistic uncertainty estimates, which is crucial in real-world applications and particularly in our information driven approach. In chapter 3 we derive a model for the covariance of the visual residuals in multi-view SfM, odometry and SLAM setups. The core of our approach is the formulation of the residual covariances as a combination of geometric and photometric noise sources. And our key novel contribution is the derivation of a term modelling how local 2D patches suffer from **perspective deformation** when imaging 3D surfaces around a point. Together, these add up to an efficient and general formulation which not only improves the accuracy of both feature-based and direct methods, but can also be used to estimate more accurate measures of the state entropy and hence better founded point visibility thresholds. We validate our model with synthetic and real data and integrate it into photometric and feature-based Bundle Adjustment, improving their accuracy with a negligible overhead.

In chapter 4 we combined our two previous contributions in the formulation and implementation of **SID-SLAM**, a complete SLAM framework for RGB-D cameras. Our main contribution is a semi-direct approach that, for the first time, combines tightly and indistinctly photometric and feature-based image measurements. Additionally, SID-SLAM uses information metrics to reduce the state size with a minimal impact in the accuracy. Our evaluation on several

public datasets shows that we further improve state-of-the-art performance regarding accuracy, robustness and computational footprint in CPU real time. We release the source code with the aim of being a SLAM solution for researchers¹ and to provide a solid framework for future developments in the field.

We recorded **Minimal Texture**, a new dataset to facilitate state-of-the-art research on semi-direct SLAM, particularly: (i) a better understanding of visual uncertainties of both features and photometric approaches [FMCT22], (ii) the efficient use of all the information on the image which maximizes SLAM robustness and reduces its computational footprint [FCT20]. Our dataset consists of 16 RGB-D sequences with conceptually simple but challenging content. The dataset was recorded with a Realsense D435i, capturing intensity and depth images of resolution 1920×1080 at rate of 30 Hz. We used a ceiling-mounted Vicon system to record millimeter-level ground truth for the camera pose.

Finally, in chapter 5 we present **DOT** (Dynamic Object Tracking) [BFC⁺21], a front-end that combines instance segmentation and multi-view geometry to generate masks for dynamic objects in order to allow SLAM systems based on rigid scene models to avoid such image areas in their optimizations. To determine which objects are actually moving, DOT segments first instances of potentially dynamic objects and then, with the estimated camera motion, tracks such objects by minimizing the photometric reprojection error. In the end, only actually dynamic masks are generated. Our results show that our approach improves significantly the accuracy and robustness of ORB-SLAM 2, especially in highly dynamic scenes. This contribution was issued in collaboration with the first author in the frame of her master thesis. Our contributions are: (i) the code is built over the classes and methods from Information-Driven Navigation implementations, (ii) the information criteria to tackle the movement decision is based on our information contributions, (iii) the experiments, results and the publication itself was carried out in collaboration with the main author.

6.2 Discussion and Future Work

There are several lines of research that build on and could improve the results of this work. The following lines of research are specially promising:

- Our “Model for Multi-View Residual Covariances based on Perspective Deformation” can be applied to different vision related problems. For example the accuracy of a camera calibration can be modelled as the trade-off between a sufficiently informative geometric configuration and the image noise that perspective deformations produce. In other tasks such as semantic segmentation or object/place recognition, perspective deformation is also an issue if viewpoints vary significantly.
- The development of a probabilistic photometric model could improve the accuracy of the information metrics. If one takes a step forward, in order to reduce SLAM computational footprint to its minimum, uncertainties that arise from noise sources such “dynamic objects” or “illumination changes” need to be tackled. We required novel algorithms and tools that estimate not only uncertainties from geometric sources (such as an RGB-D sensor) but also those associated to more complex scene behaviours.
- Efficient selection of points, keyframe insertion and elimination of redundancy in the context of information-driven SLAM can be further investigated, for example by searching for an optimal solution instead of sequential and loosely coupled point selection. The windowed keyframe optimization framework employs effective heuristics which allow the real-time operation of SLAM systems. We think that further analysis on the information of the optimization setup could offer even better results.

6.3 *Resumen de los logros de la tesis*

En esta tesis, *Information-Driven Navigation*, desarrollamos nuevos métodos probabilísticos para mejorar la precisión, robustez y eficiencia del SLAM Visual. El núcleo de nuestro trabajo y contribuciones se encuentra publicado en tres artículos principales contenidos en los capítulos 2, 3 y 4. A lo largo de estas publicaciones hemos analizado en profundidad técnicas anteriores y hemos presentado un número significativo de técnicas nuevas que han sido contrastadas y comparadas con el estado del arte. Las detalladas descripciones de nuestras soluciones del estado del arte, nuestro código libre **SID-SLAM** y el nuevo *Minimal Texture dataset* proveen una base sólida para futuros desarrollos en el campo.

De forma general, en nuestra primera publicación [FCT20] aplicamos métricas de información en una odometría visual RGB-D basada en métodos directos con el objetivo de reducir su impacto computacional. De los resultados y experimentos llevados a cabo en este trabajo, adjugamos que un mejor modelo de incertidumbre para las medidas visuales conduciría a medidas de información más realistas y se convertiría en un elemento instrumental de nuestro trabajo dirigido por información. En nuestra segunda publicación [FMCT22] derivamos un modelo de covarianzas para residuos multivista del que el elemento clave es la formulación de un término basado en la deformación perspectiva. Finalmente, combinamos ambas contribuciones en la formulación del primer sistema completo semi-directo RGB-D de SLAM que utiliza de forma integrada e indistinta características y métodos directos. A continuación desarrollamos estas tres contribuciones en más detalle junto con nuestro nuevo *Minimal Texture dataset* y la publicación de *Direct Object Tracking* (DOT) [BFC⁺21].

Como se menciona anteriormente un elemento importante de nuestras soluciones para el SLAM Visual es una **estrategia de selección de puntos basada en Teoría de la Información**. En el capítulo 2 proponemos un criterio nuevo para seleccionar los puntos más informativos para ser utilizados en una odometría RGB-D, manteniendo una precisión similar al estado del arte. Específicamente, nuestros resultados experimentales muestran que utilizar los 24 puntos más informativos es suficiente para alcanzar los resultados del estado del arte a la vez que se reduce el coste computacional hasta un factor de 10 veces. Hasta donde sabemos, esta es la primera vez que la Teoría de la Información se aplica

en métodos de odometría y SLAM directos. Creemos que nuestros resultados facilitarán el uso de la odometría visual y del SLAM en pequeñas plataformas robóticas y gafas AR/VR, que están limitadas en capacidad computacional y potencia.

Un modelo mejor de covarianzas de los residuos mejoraría la precisión de la estructura y movimiento multi-vista. Más aún, llevaría a estimaciones más realistas de la incertidumbre, que son cruciales en aplicaciones reales y particularmente en nuestra estrategia dirigida por información. En el capítulo 3 derivamos un modelo para las covarianzas de los residuos visuales para algoritmos de SfM multi-vista, odometría y SLAM. El núcleo de nuestra propuesta es la formulación de las covarianzas de los residuos como una combinación de fuentes de ruido geométricas y fotométricas. Y nuestra novedosa contribución clave es la derivación de un término que modela cómo superficies locales 2D adolecen de **deformación perspectiva** cuando se proyectan superficies 3D alrededor del punto. Todo ello, conduce a una formulación general y eficiente que no sólo mejora la precisión tanto de métodos basados en características como métodos directos, sino que también puede utilizarse para estimar medidas más precisas de la entropía del estado y por ello mejores umbrales para la visibilidad de los puntos. Validamos nuestro modelo con datos sintéticos y reales y lo integramos en un algoritmo de *Bundle-Adjustment* fotométrico y basado en características, mejorando su precisión con una sobrecarga computacional despreciable.

En el capítulo 4 combinamos las dos contribuciones anteriores para presentar **SID-SLAM**, un sistema completo de SLAM para cámaras RGB-D. Nuestra contribución principal es una estrategia semi-directa que, por primera vez, combina de forma integrada e indistinta medidas en la imagen fotométricas y basadas en características. Además, SID-SLAM utiliza métricas de información para reducir el tamaño del estado con un impacto mínimo en la precisión. Nuestra evaluación en diversos *datasets* públicos muestran que mejoramos el funcionamiento del estado del arte en cuanto a precisión, robustez y carga computacional en una CPU en tiempo real. Liberamos el código con el objetivo de convertirlo en una solución de SLAM para investigadores y para proveer un marco sólido para futuras mejoras en el campo.

Grabamos **Minimal Texture**, un *dataset* nuevo para facilitar la investigación del estado del arte en SLAM semi-directo, particularmente: (i) un mejor entendimiento de las incertidumbres visuales tanto de características como de métodos fotométricos [FMCT22], (ii) la utilización eficiente de toda la información de la imagen que maximice la robustez

del SLAM y reduzca su carga computacional [FCT20]. Nuestro *dataset* consiste en 16 secuencias RGB-D con contenido conceptualmente simple pero desafiante. El *dataset* fue grabado con una cámara Realsense D435i, capturando imágenes de intensidad y de profundidad con resolución 1920×1080 y con una frecuencia de 30 Hz. Utilizamos un sistema *Vicon* montado en el techo para registrar con precisión de milímetros un *ground truth* para la posición de la cámara.

Finalmente, en el capítulo 5 presentamos **DOT** (*Dynamic Object Tracking*) [BFC⁺21], un *front-end* que combina segmentación de instancias y geometría multi-vista para generar máscaras para objetos dinámicos que permitan a los sistemas de SLAM basados en modelos rígidos de la escena evitar ese tipo de áreas de las imágenes en sus optimizaciones. Para determinar qué objetos se están moviendo realmente, DOT segmenta primero instancias de objetos potencialmente dinámicos y después, con la estimación del movimiento de la cámara, localiza esos objetos minimizando el error fotométrico de reproyección. Al final, sólo se generan máscaras para los objetos que ciertamente se mueven. Nuestros resultados muestran que nuestro método mejora significativamente la precisión y la robustez de ORB-SLAM2, especialmente en escenas altamente dinámicas. Esta contribución fue publicada en colaboración con la primera autora en el marco de su tesis final de máster. Las contribuciones aportadas por *Information-Driven Navigation*: (i) el código se construyó sobre las clases y métodos de la odometría visual fotométrica en [FCT20], (ii) el criterio de información que toma la decisión del movimiento de los objetos está basado en las contribuciones de información de esta tesis, (iii) y los experimentos, resultados y la publicación misma fueron desarrollados en colaboración con la autora principal.

Bibliography

- [APSL08] Josep Aulinas, Yvan Petillot, Joaquim Salvi, and Xavier Lladó. The SLAM problem: a survey. *Artificial Intelligence Research and Development*, pages 363–371, 2008. [2](#), [3](#), [66](#)
- [ASC20] Charlotte Arndt, Reza Sabzevari, and Javier Civera. From points to planes—adding planar constraints to monocular SLAM factor graphs. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4917–4922. IEEE, 2020. [53](#)
- [BFC⁺21] Irene Ballester, Alejandro Fontan, Javier Civera, Klaus H Strobl, and Rudolph Triebel. DOT: Dynamic object tracking for visual SLAM. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11705–11711. IEEE, 2021. [3](#), [7](#), [10](#), [89](#), [90](#), [92](#), [94](#)
- [BFCN18] Berta Bescos, José M Fácil, Javier Civera, and José Neira. DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018. [3](#), [7](#), [72](#)
- [BLPG18] Ioan Andrei Barsan, Peidong Liu, Marc Pollefeys, and Andreas Geiger. Robust Dense Mapping for Large-Scale Dynamic Environments. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018. [72](#)
- [BM13] Jonathan T Barron and Jitendra Malik. Intrinsic scene properties from a single RGB-D image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17–24, 2013. [39](#)

- [BWC18] P. Bergmann, R. Wang, and D. Cremers. Online Photometric Calibration of Auto Exposure Video for Realtime Visual Odometry and SLAM. *IEEE Robotics and Automation Letters (RA-L)*, 3:627–634, April 2018. [3](#), [6](#)
- [CC15] Alejo Concha and Javier Civera. An evaluation of robust cost functions for RGB direct mapping. In *2015 European Conference on Mobile Robots (ECMR)*, pages 1–8. IEEE, 2015. [3](#), [16](#)
- [CC17] Alejo Concha and Javier Civera. RGBDTAM: A cost-effective and accurate RGB-D tracking and mapping system. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 6756–6763. IEEE, 2017. [3](#), [4](#), [7](#), [15](#), [17](#), [20](#), [22](#), [39](#), [53](#), [62](#), [64](#)
- [CCC⁺16] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John 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. [2](#), [3](#), [51](#), [70](#)
- [CCGFO21] Joan P Company-Corcoles, Emilio Garcia-Fidalgo, and Alberto Ortiz. MSC-VO: Exploiting Manhattan and Structural Constraints for Visual Odometry. *arXiv preprint arXiv:2111.03408*, 2021. [53](#)
- [CD08] Margarita Chli and Andrew J Davison. Active matching. In *European conference on computer vision*, pages 72–85. Springer, 2008. [14](#), [18](#), [33](#)
- [CD09] Margarita Chli and Andrew J Davison. Active matching for visual tracking. *Robotics and Autonomous Systems*, 57(12):1173–1187, 2009. [3](#), [5](#), [14](#)
- [CDM08] Javier Civera, Andrew J Davison, and JM Martinez Montiel. Inverse depth parametrization for monocular SLAM. *IEEE transactions on robotics*, 24(5):932–945, 2008. [3](#)
- [CER⁺21] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Transactions on Robotics*, 2021. [3](#), [54](#), [62](#), [64](#), [65](#), [67](#)

- [CICD15] Siddharth Choudhary, Vadim Indelman, Henrik I Christensen, and Frank Dellaert. Information-based reduced landmark SLAM. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4620–4627, 2015. 15
- [CKB⁺14] Luca Carlone, Zsolt Kira, Chris Beall, Vadim Indelman, and Frank Dellaert. Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4290–4297. IEEE, 2014. 14
- [CMH20] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2, 2020. 81, 87
- [CZK15] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust Reconstruction of Indoor Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3, 9, 38
- [Dav05] Andrew J Davison. Active search for real-time vision. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 66–73. IEEE, 2005. 3, 5, 14, 18
- [Dav18] Andrew J Davison. FutureMapping: The computational structure of spatial AI systems. *arXiv preprint arXiv:1803.11288*, 2018. 2, 3, 13
- [DNZ⁺17] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundl fusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(4):1, 2017. 3, 62, 64, 65
- [DRMS07] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007. 2, 3
- [DSCU] N Demmel, C Sommer, D Cremers, and V Usenko. Square root bundle adjustment for large-scale reconstruction. 3, 5
- [DSS⁺21] N Demmel, D Schubert, C Sommer, D Cremers, and V Usenko. Square Root Marginalization for Sliding-Window Bundle Adjustment. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 3, 5

- [EHE⁺12] Felix Endres, Jürgen Hess, Nikolas Engelhard, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. An evaluation of the RGB-D SLAM system. In *2012 IEEE International Conference on Robotics and Automation*, pages 1691–1696. IEEE, 2012. [3](#), [64](#)
- [EHS⁺13] Felix Endres, Jürgen Hess, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. 3-D mapping with an RGB-D camera. *IEEE transactions on robotics*, 30(1):177–187, 2013. [3](#), [34](#), [64](#)
- [EKC17] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017. [3](#), [5](#), [6](#), [7](#), [13](#), [15](#), [16](#), [20](#), [22](#), [23](#), [24](#), [25](#), [32](#), [34](#), [36](#), [39](#), [40](#), [41](#), [43](#), [46](#), [53](#), [54](#), [55](#), [60](#), [71](#), [74](#), [76](#)
- [ESC13] Jakob Engel, Jürgen Sturm, and Daniel Cremers. Semi-dense visual odometry for a monocular camera. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1456, 2013. [3](#), [5](#)
- [ESC14] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *European conference on computer vision*, pages 834–849. Springer, 2014. [3](#), [5](#), [53](#)
- [EUC16] Jakob Engel, Vladyslav Usenko, and Daniel Cremers. A photometrically calibrated benchmark for monocular visual odometry. *arXiv preprint arXiv:1607.02555*, 2016. [3](#), [6](#), [40](#)
- [FCT20] Alejandro Fontan, Javier Civera, and Rudolph Triebel. Information-driven direct rgb-d odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4929–4937, 2020. [2](#), [3](#), [6](#), [10](#), [33](#), [47](#), [48](#), [49](#), [56](#), [66](#), [88](#), [90](#), [92](#), [94](#)
- [FMCT22] Alejandro Fontan, Laura Oliva Maza, Javier Civera, and Rudolph Triebel. A Model for Multi-View Residual Covariances based on Perspective Deformation. *IEEE Robotics and Automation Letters*, 2022. [2](#), [3](#), [6](#), [10](#), [55](#), [66](#), [88](#), [90](#), [92](#), [93](#)

- [FPS14] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 15–22. IEEE, 2014. [3](#), [7](#), [39](#), [53](#)
- [FZG⁺16] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. SVO: Semidirect visual odometry for monocular and multi-camera systems. *IEEE Transactions on Robotics*, 33(2):249–265, 2016. [3](#), [7](#), [53](#), [62](#)
- [GBN08] Pierre Fite Georgel, Selim Benhimane, and Nassir Navab. A Unified Approach Combining Photometric and Geometric Information for Pose Estimation. In *BMVC*, pages 1–10. Citeseer, 2008. [54](#)
- [GLT12] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012. [53](#)
- [GLU12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. [70](#), [81](#), [87](#)
- [GOMGJ17] Ruben Gomez-Ojeda, Francisco-Angel Moreno, and Javier Gonzalez-Jimenez. Accurate stereo visual odometry with gamma distributions. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1423–1428. IEEE, 2017. [58](#)
- [GOMZN⁺19] Ruben Gomez-Ojeda, Francisco-Angel Moreno, David Zuniga-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*, 35(3):734–746, 2019. [53](#)
- [GSM19] Sourav Garg, Niko Suenderhauf, and Michael Milford. Semantic-geometric visual place recognition: a new perspective for reconciling opposing views. *The International Journal of Robotics Research*, page 0278364919839761, 2019. [32](#)

- [GWCV16] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2016. [81](#), [87](#)
- [GWDC18] Xiang Gao, Rui Wang, Nikolaus Demmel, and Daniel Cremers. LDSO: Direct sparse odometry with loop closure. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2198–2204. IEEE, 2018. [3](#), [7](#), [53](#)
- [HCMH16] Wajahat Hussain, Javier Civera, Luis Montano, and Martial Hebert. Dealing with small data and training blind spots in the Manhattan world. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016. [32](#)
- [HCSD10] Ankur Handa, Margarita Chli, Hauke Strasdat, and Andrew J Davison. Scalable active matching. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1546–1553. IEEE, 2010. [14](#)
- [HGDG17] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. [72](#)
- [HHW⁺18] Jerry Hsiung, Ming Hsiao, Eric Westman, Rafael Valencia, and Michael Kaess. Information sparsification in visual-inertial odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1146–1153. IEEE, 2018. [14](#)
- [HKL13] Guoquan Huang, Michael Kaess, and John J Leonard. Consistent sparsification for graph optimization. In *2013 European Conference on Mobile Robots*, pages 150–157, 2013. [14](#)
- [HWMD14] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *2014 IEEE international conference on Robotics and automation (ICRA)*, pages 1524–1531. IEEE, 2014. [3](#), [9](#), [39](#), [42](#), [43](#), [64](#), [65](#)

- [HYMH20] Jiahui Huang, Sheng Yang, Tai-Jiang Mu, and Shi-Min Hu. ClusterVO: Clustering Moving Instances and Estimating Visual Odometry for Self and Surroundings, 2020. 81
- [HYZ⁺19] Jiahui Huang, Sheng Yang, Zishuo Zhao, Yu-Kun Lai, and Shi-Min Hu. Cluster-SLAM: A SLAM Backend for Simultaneous Rigid Body Clustering and Motion Estimation. 2019. 72
- [HZ03] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 75
- [IA99] Michal Irani and P Anandan. About direct methods. In *International Workshop on Vision Algorithms*, pages 267–277. Springer, 1999. 3, 52
- [IPAC09] Viorela Ila, Josep M Porta, and Juan Andrade-Cetto. Information-based compact pose SLAM. *IEEE Transactions on Robotics*, 26(1):78–93, 2009. 14
- [IZN⁺16] Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. VolumeDeform: Real-time Volumetric Non-rigid Reconstruction. October 2016. 71
- [Kae15] Michael Kaess. Simultaneous localization and mapping with infinite planes. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4605–4611. IEEE, 2015. 53
- [Kho11] Kouros Khoshelham. Accuracy analysis of kinect depth data. In *ISPRS workshop laser scanning*, volume 38, 2011. 39
- [KM07] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007. 3, 34
- [KMZS20] Juichung Kuo, Manasi Muglikar, Zichao Zhang, and Davide Scaramuzza. Redesigning SLAM for arbitrary multi-camera systems. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2116–2122. IEEE, 2020. 33

- [KS12] Henrik Kretschmar and Cyrill Stachniss. Information-theoretic compression of pose graphs for laser-based SLAM. *The International Journal of Robotics Research*, 31(11):1219–1230, 2012. 15
- [KSC13a] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense visual SLAM for RGB-D cameras. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2100–2106. IEEE, 2013. 3, 4, 15, 16, 23, 26, 29, 33, 53, 54, 58, 64, 65
- [KSC13b] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Robust odometry estimation for RGB-D cameras. In *2013 IEEE international conference on robotics and automation*, pages 3748–3754. IEEE, 2013. 3, 4, 15, 16, 21, 22
- [LC18] Seong Hun Lee and Javier Civera. Loosely-coupled semi-direct monocular slam. *IEEE Robotics and Automation Letters*, 4(2):399–406, 2018. 3, 7, 53
- [LMB⁺14] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context, 2014. 74
- [Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 36
- [LPBM19] Jose Lamarca, Shaifali Parashar, Adrien Bartoli, and JMM Montiel. Defslam: Tracking and mapping of deforming scenes from monocular sequences. *arXiv preprint arXiv:1908.08918*, 2019. 71
- [MAMT15] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 3, 5, 6, 7, 13, 22, 54, 70
- [MAT17a] Raul Mur-Artal and Juan 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. 3, 9, 13, 23, 26, 29, 32, 34, 36, 41, 46, 47, 49, 50, 53, 62, 64, 65, 67

- [MAT17b] Raúl Mur-Artal and Juan 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. 69, 70, 72, 80
- [Mat19] Inc. Matterport. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow, 2019. URL: https://github.com/matterport/Mask_RCNN [Online. Accedido el 03/12/2019]. 74
- [MDR04] Nicholas Molton, Andrew J Davison, and Ian Reid. Locally Planar Patch Features for Real-Time Structure from Motion. In *Bmvc*, pages 1–10, 2004. 3, 6, 34, 35, 42
- [MKSC16] Lingni Ma, Christian Kerl, Jörg Stückler, and Daniel Cremers. CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1285–1291. IEEE, 2016. 53
- [MSFV⁺21] Lukas Meyer, Michal Smíšek, Alejandro Fontan Villacampa, Laura Oliva Maza, Daniel Medina, Martin J Schuster, Florian Steidle, Mallikarjuna Vayugundla, Marcus G Müller, Bernhard Rebele, et al. The MADMAX data set for visual-inertial rover navigation on Mars. *Journal of Field Robotics*, 2021. 3, 9, 53, 67
- [NFS15] Richard A. Newcombe, Dieter Fox, and Steven M. Seitz. DynamicFusion: Reconstruction and Tracking of Non-Rigid Scenes in Real-Time. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 71
- [NIH⁺11] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. IEEE, 2011. 3, 4, 64

- [NLD11] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. DTAM: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011. [3](#), [4](#)
- [NZIS13] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3D reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)*, 32(6):1–11, 2013. [3](#), [64](#)
- [PDL17] Lukas Platinisky, Andrew J Davison, and Stefan Leutenegger. Monocular visual odometry: Sparse joint optimisation or dense alternation? In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5126–5133. IEEE, 2017. [3](#), [5](#), [60](#)
- [PFC⁺15] Taihú Pire, Thomas Fischer, Javier Civera, Pablo De Cristóforis, and Julio Jacobo Berles. Stereo parallel tracking and mapping for robot localization. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1373–1378. IEEE, 2015. [13](#)
- [PS19] Songyou Peng and Peter Sturm. Calibration Wizard: A guidance system for camera calibration based on modelling geometric and corner uncertainty. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1497–1505, 2019. [3](#), [6](#), [32](#), [33](#), [34](#)
- [PVA⁺17] Albert Pumarola, Alexander Vakhitov, Antonio Agudo, Alberto Sanfeliu, and Francese Moreno-Noguer. PL-SLAM: Real-time monocular visual SLAM with points and lines. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 4503–4508. IEEE, 2017. [53](#)
- [PZK17] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Colored point cloud registration revisited. In *Proceedings of the IEEE international conference on computer vision*, pages 143–152, 2017. [54](#)
- [QLS18] Tong Qin, Peiliang Li, and Shaojie Shen. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018. [9](#), [13](#)

- [RACC20] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2020. 34
- [RBA18] Martin Rünz, Maud Buffier, and Lourdes Agapito. MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects, 2018. 72
- [RRKB11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011. 36, 46
- [SB12] Jörg Stückler and Sven Behnke. Integrating depth and color cues for dense multi-resolution scene mapping using rgb-d cameras. In *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 162–167. IEEE, 2012. 3, 53, 64
- [SC19] Patrik Schmuck and Margarita Chli. On the Redundancy Detection in Keyframe-based SLAM. In *2019 International Conference on 3D Vision (3DV)*, pages 594–603, 2019. 3, 18, 25
- [SDU⁺18] David Schubert, Nikolaus Demmel, Vladyslav Usenko, Jorg Stuckler, and Daniel Cremers. Direct sparse odometry with rolling shutter. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 682–697, 2018. 3, 6
- [SEE⁺12a] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A Benchmark for the Evaluation of RGB-D SLAM Systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012. 3, 7, 25, 33, 44, 45, 47, 48, 49, 56, 62, 64
- [SEE⁺12b] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580. IEEE, 2012. 82

- [SF16] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 34, 36
- [SG18] Dominik Schlegel and Giorgio Grisetti. HBST: A hamming distance embedding binary search tree for feature-based visual place recognition. *IEEE Robotics and Automation Letters*, 3(4):3741–3748, 2018. 61
- [Sha48] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948. 2
- [SJP⁺] Raluca Scona, Mariano Jaimez, Yvan R. Petillot, Maurice Fallon, and Daniel Cremers. StaticFusion: Background Reconstruction for Dense RGB-D SLAM in Dynamic Environments. In *2018 ICRA*. IEEE. 72
- [SMD10] Hauke Strasdat, JMM Montiel, and Andrew J Davison. Real-time monocular SLAM: Why filter? In *2010 IEEE International Conference on Robotics and Automation*, pages 2657–2664. IEEE, 2010. 3, 5, 33, 56
- [SMD12] Hauke Strasdat, José MM Montiel, and Andrew J Davison. Visual SLAM: why filter? *Image and Vision Computing*, 30(2):65–77, 2012. 14, 18
- [SSC11] Frank Steinbrücker, Jürgen Sturm, and Daniel Cremers. Real-time visual odometry from dense RGB-D images. In *2011 IEEE international conference on computer vision workshops (ICCV Workshops)*, pages 719–722. IEEE, 2011. 3, 4
- [SSP19] Thomas Schöps, Torsten Sattler, and Marc Pollefeys. Bad slam: Bundle adjusted direct rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 134–144, 2019. 3, 4, 5, 7, 9, 52, 53, 54, 56, 60, 62, 63, 64, 65, 67
- [Str12] Hauke Strasdat. *Local accuracy and global consistency for efficient visual SLAM*. PhD thesis, Department of Computing, Imperial College London, 2012. 3, 15, 54, 75
- [SVN20] Yawar Siddiqui, Julien Valentin, and Matthias Nießner. ViewAL: Active learning with viewpoint entropy for semantic segmentation. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9433–9443, 2020. 32

- [UESC16] Vladyslav Usenko, Jakob Engel, Jörg Stückler, and Daniel Cremers. Direct visual-inertial odometry with stereo cameras. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1885–1892. IEEE, 2016. 24
- [VDWB11] John Vial, Hugh Durrant-Whyte, and Tim Bailey. Conservative sparsification for efficient and consistent approximate estimation. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 886–893, 2011. 15
- [VLF04] Luca Vacchetti, Vincent Lepetit, and Pascal Fua. Combining edge and texture information for real-time accurate 3d camera tracking. In *Third IEEE and ACM international symposium on mixed and augmented reality*, pages 48–56. IEEE, 2004. 53
- [VSUC18] Lukas Von Stumberg, Vladyslav Usenko, and Daniel Cremers. Direct sparse visual-inertial odometry using dynamic marginalization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2510–2517. IEEE, 2018. 24
- [WJK⁺13] Thomas Whelan, Hordur Johannsson, Michael Kaess, John J Leonard, and John McDonald. Robust real-time visual odometry for dense RGB-D mapping. In *2013 IEEE International Conference on Robotics and Automation*, pages 5724–5731. IEEE, 2013. 3, 4, 54
- [WKF⁺12] Thomas Whelan, Michael Kaess, Maurice Fallon, Hordur Johannsson, John Leonard, and John McDonald. Kintinuous: Spatially extended kinectfusion. 2012. 3, 4
- [WKJ⁺15] Thomas Whelan, Michael Kaess, Hordur Johannsson, Maurice Fallon, John J Leonard, and John McDonald. Real-time large-scale dense RGB-D SLAM with volumetric fusion. *The International Journal of Robotics Research*, 34(4-5):598–626, 2015. 3, 64

- [WKM⁺19] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019. 70, 74
- [WLSM⁺15] Thomas Whelan, Stefan Leutenegger, R Salas-Moreno, Ben Glocker, and Andrew Davison. ElasticFusion: Dense SLAM without a pose graph. *Robotics: Science and Systems*, 2015. 3, 4
- [WS14] Kyle Wilson and Noah Snavely. Robust global translations with 1DSfM. In *European Conference on Computer Vision*, pages 61–75, 2014. 34
- [WSC17] R. Wang, M. Schwörer, and D. Cremers. Stereo DSO: Large-Scale Direct Sparse Visual Odometry with Stereo Cameras. In *International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017. 3
- [WSMG⁺16] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. ElasticFusion: Real-time dense SLAM and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016. 3, 4, 64, 65
- [WXLH13] Yue Wang, Rong Xiong, Qianshan Li, and Shoudong Huang. Kullback-leibler divergence based graph pruning in robotic feature mapping. In *2013 European Conference on Mobile Robots*, pages 32–37. IEEE, 2013. 15
- [XLT⁺18] Binbin Xu, Wenbin Li, Dimos Tzoumanikas, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. MID-Fusion: Octree-based Object-Level Multi-Instance Dynamic SLAM, 2018. 72
- [YWGC18] Nan Yang, Rui Wang, Xiang Gao, and Daniel Cremers. Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect. *IEEE Robotics and Automation Letters*, 3(4):2878–2885, 2018. 3, 6, 34, 38, 40, 46, 52, 53
- [YYR17] Zhixin Yan, Mao Ye, and Liu Ren. Dense visual SLAM with probabilistic surfel map. *IEEE transactions on visualization and computer graphics*, 23(11):2389–2398, 2017. 3, 64

- [ZKK21] Lipu Zhou, Daniel Koppel, and Michael Kaess. LiDAR SLAM with Plane Adjustment for Indoor Environment. *IEEE Robotics and Automation Letters*, 6(4):7073–7080, 2021. 53
- [ZLK18] Yi Zhou, Hongdong Li, and Laurent Kneip. Canny-VO: Visual Odometry With RGB-D Cameras Based on Geometric 3-D–2-D Edge Alignment. *IEEE Transactions on Robotics*, 35(1):184–199, 2018. 26, 29, 53
- [ZV18] Yipu Zhao and Patricio A Vela. Good feature selection for least squares pose optimization in VO/VSLAM. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1183–1189. IEEE, 2018. 3, 6, 33
- [ZV20] Yipu Zhao and Patricio A Vela. Good feature matching: Toward accurate, robust vo/vslam with low latency. *IEEE Transactions on Robotics*, 36(3):657–675, 2020. 3, 6, 33
- [ZWK21] Lipu Zhou, Shengze Wang, and Michael Kaess. DPLVO: Direct Point-Line Monocular Visual Odometry. *IEEE Robotics and Automation Letters*, 6(4):7113–7120, 2021. 53