Visual SLAM

Enhancing Direct Visual Odometry Through the Integration of Deep Learning Approaches

Project Report

Magnus K. Gjerde

Aalborg University
Electronics and IT

Electronics and IT
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Visual SLAM: Enhancing Direct Visual Odometry Through the Integration of Deep Learning Approaches

Theme:
Computer Vision

Project Period:
Spring Semester 2023

Project Group:
1045

Participant(s):
Magnus Kaufmann Gjerde

Supervisor(s):
Kamal Nasrollahi

Copies: 1

Page Numbers: 51

Date of Completion:
June 1, 2023

Abstract:

This paper investigates the field of direct visual odometry and specifically the implementation of hybrid approaches between deep learning and classical handcrafted methods. This project introduces a new approach that integrates a deblurring module with a saliency predictor to perform better point sampling which increases trajectory estimation accuracy in blurry frames, often caused by rapid camera movements or long exposure times in dimly lit conditions. Benchmark testing against DSO and SalientDSO on the EuRoC MAV dataset demonstrated consistent improvements, with the proposed system achieving an average Absolute Trajectory Error (ATE) of 0.26m, compared to 0.335m for DSO and 0.303m for SalientDSO. Further research directions include investigating other image improvement methods such as dehazing, denoising, or image enhancement for even more accurate results.

Contents

Preface

Reading Manual

This report is written for a one-page format. It is intended to be read digitally, which may be required for some of the figures, as the reader may need the capability to zoom in to read the details.

Aalborg University, June 1, 2023

Magnus K. Gjerde

Magnus Kaufmann Gjerde
mgjerd18@student.aau.dk

Chapter 1

Introduction

The rapid development of Visual Simultaneous Localization and Mapping (V-SLAM) has facilitated significant advancements in various industries, including autonomous vehicles, robotics, and augmented reality. As a result, novel applications such as indoor mapping, robot-assisted surgery, immersive gaming experiences, and planetary exploration, have emerged, demonstrating the potential of V-SLAM technology [1, 2, 3, 4, 5].
V-SLAM has been in development since the 2000s [6] leading to a vast selection of methods within the research community. Each method is designed to tackle unique challenges and environments. Specific V-SLAM methods have been made to deliver more detailed maps, facilitate mapping during fast motion, manage large environments, or cope with dynamic scenes as highlighted in Kazerouni et al.'s survey [7]. Some specific examples of V-SLAM applications are:

1. Disaster response: Deploying autonomous robots or drones to assess the damage, locate survivors, and navigate through hazardous environments during natural disasters, such as earthquakes or tsunamis.

2. Environmental monitoring and conservation: Mapping and localization to monitor and map forest ecosystems, enabling detection of illegal logging, monitoring climate change, or wildlife tracking for conservation purposes.

3. Infrastructure inspection and maintenance: Utilizing V-SLAM technology for inspection and maintenance of critical infrastructure such as bridges, tunnels, and power plants.

As an example, in the ongoing fight against climate change and the need for conserving biodiversity, the scientific community increasingly relies on advanced technologies to understand and manage natural ecosystems. One such technological frontier is the use of robotics for environmental monitoring and conservation. Autonomous robotic systems can be deployed to monitor and map vast forest ecosystems, track indicators of illegal logging, observe the impact of climate change, and even follow wildlife movements and patterns. V-SLAM plays a critical role in these applications and is one of the most efficient methods for building 3D maps and

estimating the position of a robot within it [7]. Building on the contextual understanding of V-SLAM technology, the forward-thinking organization, Carnegie Robotics which has supplied one of their high-end rugged stereo cameras for this project will be introduced next.

1.1 Company collaboration: Carnegie Robotics

The company Carnegie Robotics is a Pittsburgh-based company that specializes in autonomy in rugged environments. It was founded in 2010 by a team of experts in robotics, computer vision, electrical engineering, and mechanical engineering. The company has since grown to employ over 150 people. Their project portfolio continuously expands, and today they are producing solutions within the fields of, autonomous mining, infrastructure inspection, agricultural automation, defense robotics, and delivering sensing and perception systems for autonomous boats. A commonality for the aforementioned fields is that the environments are mostly large and unknown. The rugged and unpredictable nature of the environments makes it a challenge for Carnegie Robotics' autonomous solutions to operate. Unlike autonomous cars that can rely on pre-existing road maps and traffic patterns. Therefore Carnegie Robotics specializes in developing robust and advanced sensing and control systems that enable their robots to navigate challenging environments without relying on prior knowledge. To further improve the performance of their autonomous systems, it is essential for Carnegie Robotics to have advanced sensing and perception capabilities, one key technology is V-SLAM. By developing a robust V-SLAM algorithm for their MultiSense cameras, Carnegie Robotics can significantly improve the performance of their autonomous systems enabling their robots to better understand and navigate the rugged environment.

As developing a full V-SLAM project for the course of this project will be too ambitious given the level of complexity and time to complete this project, it is necessary to focus on a smaller module of V-SLAM. However, in order to do that, a fundamental understanding of the field of V-SLAM is required. The guiding question for the following chapter is stated below.

Initial Problem Formulation:

"What is the current state of the V-SLAM field and what obstacles does it encounter in real-world applications?"

The following chapter will try to answer the initial problem statement defined above.

Chapter 2

Problem Analysis

This chapter consists of two main sections. Firstly an impact analysis, that answers why visual odometry and SLAM are important for autonomy and to what extent further development will impact the real world. Secondly, it consists of a technical analysis that analyzes the current research progress and implementations, ending with a brief overview of current challenges in the field.

2.1 Real-world usage of Visual SLAM

The current state of V-SLAM has enabled new applications and advancements across various industries. It has had a significant social, economic, and commercial impact on various application areas such as autonomous vehicles, drones, robotics, and augmented reality.
In industry, advances in V-SLAM has made it possible to create accurate maps of indoor environments such as shopping malls, museum, and airports. For example navVis, a German company has utilized SLAM to create detailed 3D maps of large indoor spaces for navigation and facility management. [1]
In robot-assisted Surgery, the da Vinci Surgical System utilizes a stereo vision system with V-SLAM capabilities to provide the surgeon with accurate 3D visualization during procedures. The robot-assisted surgery helped the surgeon perform invasive procedures with less damage to the patient which led to faster recovery time. [2]
The launch of application programming interfaces (API) for mobile devices: ARKit and ARCore giving iOS and Android developers new opportunities to create immersive applications and games. While these APIs are proprietary, they heavily depend on SLAM technology to perform the localization and mapping for Virtual and Augmented reality [4]. Similarly, developers using Microsoft's Hololens have created a real-life-sized interactive Super Mario Smash game playable outdoors [3].
Improvements in SLAM have made automatic inventory management possible and are often present in large warehouses. Amazon Robotics (formerly Kiva Systems) set an example for using robotics to automate operations in warehouses. While such a system does not solely rely

on a camera it is often included as an input to a more extensive SLAM system.[8]

It is clear that improvements in the V-SLAM field have affected several areas making autonomy possible in areas that were previously inaccessible to robotics. As a result, we see commercial applications such as the ones mentioned above. However, to keep improving the technology there has to be an incentive to do so. Therefore next section will explore what the future might bring if the V-SLAM technology is improved.

2.1.1 Future possibilities for Visual SLAM

As robotics and SLAM keeps developing, new possibilities for automation arise. In research, we see development in advanced robotic telepresence. This could involve remotely controlled robots capable of performing highly precise tasks, such as repair and maintenance of satellites or space stations, without the need for human presence on-site. In February 2023, a group from German Aerospace Center envisioned and proposed a system for an aerial manipulator with teleoperation. The drone carrying the manipulator used lidar-based SLAM to localize itself according to the object to be manipulated. The human operator could control the aerial manipulator safely from the ground while receiving visual and haptic feedback from the robot.[9] Future research in this area could result in products that will eventually help workers perform high-risk tasks safely from a remote location.
Swarm robotics is another hot topic in research. As V-SLAM continues to improve, swarm robotics could also be employed in unknown environments. This could prove helpful in areas such as disaster response, environmental monitoring for climate change, or on search and rescue missions.
In the medical field, inspired by the da Vinci Surgery System, robot-assisted microsurgery may be a possibility if localization and mapping technology is improved. Highly precise procedures such as cell manipulation or tissue engineering would be a possibility as the accuracy and reliability of V-SLAM and robotics continue to improve.[10]

It is clear that continuous improvement in robot localization and mapping will be fundamental to improving autonomy in vastly different areas. The next section will describe a high-level overview of SLAM and V-SLAM technology.

2.2 SLAM

For a mobile robot to perform some tasks by itself, it needs to be able to adapt to an unfamiliar environment. To perform either searching, fetching, or coverage tasks the robot requires some sense of localization and position. SLAM refers to the process by which a mobile robot adapts to an unfamiliar environment to perform tasks such as searching, fetching, or coverage. Since its proposal in 1986, SLAM has received extensive attention in combination with different research fields such as robotics, virtual reality, autonomous driving, and drones. A subcategory of

SLAM is V-SLAM. The term visual is added when a camera is the main sensor element to perform the localization and mapping. A SLAM system typically deploys two parts. The two parts are depicted in Figure 2.1 [11]. Firstly the front end provides pose estimates for the local trajectory. The other part usually called the back end optimizes the pose estimates from the front end trying to achieve global consistency. In a V-SLAM system, the pose estimated from the front end is estimated by consecutive images from a camera.

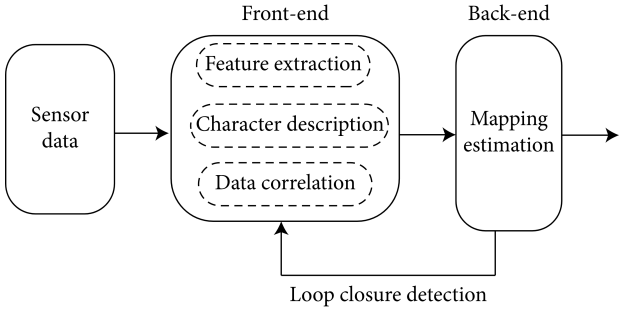


Figure 2.1: Illustration of the Frontend and backend architecture in SLAM. [11]

2.2.1 Frontend

The front end itself in a V-SLAM framework can be subcategorized further into the research field of visual odometry. A visual odometry system is typically only concerned with the local trajectory over a set of the latest few keyframes. Therefore a visual odometry system will always drift, meaning errors will accumulate and result in a growing inaccurate trajectory, however, approaches to minimize this drift have been successful and have further helped full V-SLAM systems in estimating a more accurate global trajectory. The majority of visual odometry methods are based on error minimization algorithms when attempting to estimate motion between camera frames. Within error minimization methods, the feature point method and the photometric method are two of the most popular used visual odometry solutions. The feature point technique minimizes a geometric error when reprojecting the world coordinates of tracked features, while the photometric method attempts to minimize a photometric error, i.e. the pose is recovered based on intensity changes from a set of recent images. Another name of the distinction is indirect and direct methods respectively.[11]

2.2.2 Backend

As stated earlier, the backend is responsible for optimizing the frontend information and producing a complete map that is globally coherent. To achieve this the backend is typically composed of three primary components: optimization module, loop detection, and loop closure. Firstly, the optimization module receives the poses from the front end and is responsible for refining them to generate a globally consistent map. A widely used method is graph-based optimization and is seen in popular state-of-the-art V-SLAM methods such as ORB-SLAM [12],

where poses and landmarks are represented as nodes and the transformation between them is interpreted as edges. When landmarks are observed from different poses, the pose graph can then be optimized using optimization techniques such as Gauss-Newton which aim to adjust the positions of the poses and landmarks to achieve a more globally coherent map.[13]
Secondly, the loop detection module is responsible for identifying when the robot has revisited a previously explored area. A popular technique to do this is called Bag-of-Words and is also used in ORB-SLAM [12]. This is essential to correct the accumulated drift in the front end. When a loop is detected, the pose graph can be updated such that one node can have multiple edges. After this, the new pose graph can then be optimized in the next module.
Thirdly, the loop closure module attempts to update the pose graph by adding constraints that enforce consistency between the current pose and previously visited locations. Loop closure optimization is also typically performed using optimization techniques like the aforementioned Gauss-Newton method.[13]

This project will focus on the frontend part of a V-SLAM system. Due to the relatively short time span for the thesis project, it would not be feasible to consider a whole V-SLAM system in detail. Since the front end is a fundamental part of the system, and improvements in this area will subsequently improve the performance of V-SLAM systems, this project will investigate visual odometry and its challenges in producing accurate pose estimations.

2.3 Related work for visual odometry

This section will summarize the current research development of visual odometry. In Table 2.1 is a complete list of related works that are investigated in this section. They are categorized based on their approaches to visual odometry which can either be direct, indirect, deep learning, or hybrid methods. Hybrid methods are a combination of methods from both traditional (direct/indirect) approaches and deep learning methods. For example, utilizing a trained neural network to perform place recognition in a loop detection module.

Category	Names
Direct Methods	**DSO:** Direct Sparse Odometry **DM-VIO:** Delayed Marginalization Visual-Inertial Odometry **VI-DSO:** Direct Visual-Inertial Sparse Odometry
Indirect and LKT Methods	**BASALT:** Visual-Inertial Mapping with Non-Linear Factor Recovery **VINS-Motion:** A Robust and Versatile Monocular Visual-Inertial State Estimator
Deep Learning Methods	**DPVO:** Deep Patch visual odometry **DEEPVO:** End to end visual odometry using deep learning **ContextAVO:** Local context guided and refining poses for deep visual odometry
Hybrid methods	**SalientDSO:** Direct Sparse Odometry with scene segmentation and saliency maps **MBA-VO:**Motion Blur Aware visual odometry **DeblurSLAM:** A Novel Visual SLAM System Robust in Blurring Scene

Table 2.1: List of works included in the related work analysis. They are categorized by their main approach to solving the visual odometry problem.

2.3.1 Direct methods

Direct visual odometry estimates the motion of a camera by directly analyzing the intensity values of the pixels between consecutive frames. The following section presents the state-of-the-art Direct visual odometry methods.

DM-VIO, VI-DSO and DSO

In 2022, Delayed Marginalization Visual-Intertial Odometry (DM-VIO) was presented [14]. It is a real-time monocular odometry system based on previous research from several papers but most notably the 2016 Direct Sparse Odometry (DSO) paper [15], and the 2018 Visual Intertial Direct Sparse Odometry (VI-DSO) paper [16]. VI-DSO is again an extension of DSO (Direct Sparse Odometry) where inertial information was added to the visual odometry to minimize the error introduced by rapid movements, pure rotations, and bad image quality. DM-VIO employs two novel techniques namely delayed marginalization and their coined term pose graph bundle adjustment (PGBA). PGBA is a bundle adjustment-like procedure to optimize a pose graph for a local trajectory. The novel techniques are a step in addressing the shortcomings of the IMU integration in VI-DSO. DM-VIO then presents a tightly coupled implementation of visual-inertial Odometry where the energy function jointly optimizes for IMU and visual parameters. As a result, DM-VIO is the state-of-the-art implementation for visual odometry

methods and its performance is even comparable to that of ORBSLAM3 which is a full VI-SLAM system [14]. Most of the achievements stem from the publication of the direct image formulation model from DSO. By minimizing a photometric error, the optimization process does not need to adhere to feature extraction and matching which allows the system to maintain tracking accuracy in sparse and textureless areas. Figure 2.2 shows an image from the DSO running on a frame from the European Robotics Challenge (EuRoC) Micro Aerial Vehicle (MAV) dataset in a relatively textureless scene.

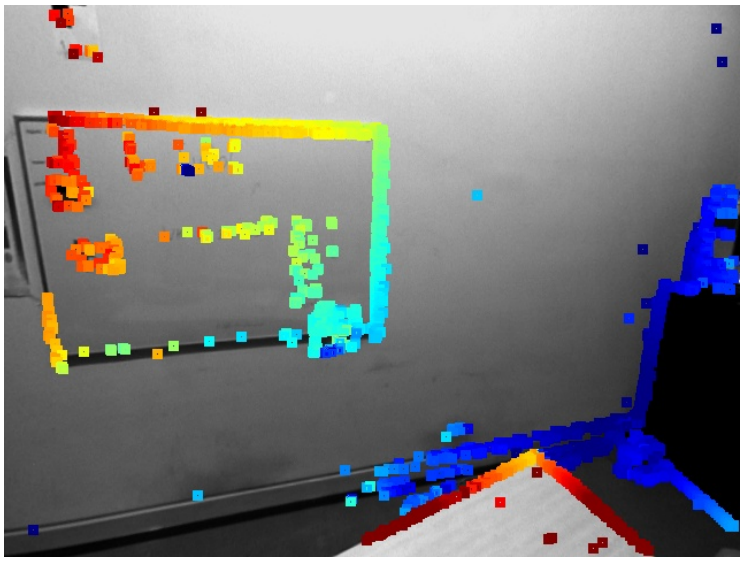


Figure 2.2: DSO running on the EuRoC MAV dataset. This frame illustrates the direct formulation's ability to track when the image contains little texture

In the Figure 2.2, the colors of the points is determined by the depth value assigned to each point which is sampled from the jet color map.

2.3.2 Indirect and optical flow methods

Lately in Indirect methods development, besides from ORB-SLAM3, feature descriptors such as ORB and SIFT descriptors have been switched out with corner methods such as FAST and Harris Corner methods. The matching algorithm heavily relies on the Lukas-Kanada-Tomasi algorithm to generate sparse optical flow from a set of consecutive images. This combination along with the integration of an Inertial Measurement Unit has shown improvements in the Visual-Intertial Odometry field and pushed state-of-the-art solutions towards more accurate implementations. A commonality for indirect methods is that the features collected are more easily extended to include SLAM components such as place recognition for loop detection and loop closure. The following sections are a description of two methods in that scope. [17] [18]

BASALT

In 2020, Usenko et al persented BASALT [18]. BASALT is a tightly integrated visual-inertial odometry system. For each frame, they minimize non-linear energy that consists of reprojection terms, IMU terms, and a marginalization prior in a sliding window. Like VINS-Motion, they also use the KLT for sparse optical flow, but they use the FAST algorithm for feature detection. An example of the KLT tracks estimated by their system is shown below in Figure 2.3. The two images presented in the figure shows the tracks on the same object from two different viewpoints and slightly different exposure.

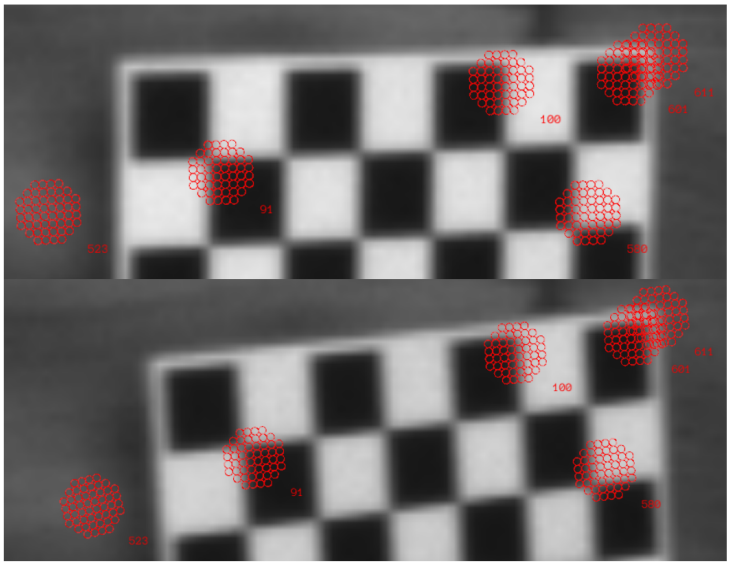


Figure 2.3: Example of KLT tracks estimated by the BASALT system. Despite changes in both viewpoint and exposure time the method is able to estimate the warp between the patches in the images

Additionally, Basalt also adds a global map optimization module, where Oriented BRIEF and Rotated Fast (ORB) features are extracted independently from the keyframes. The novelty presented in BASALT is reintegrating non-linear factors into the initially linearized optimization problem. Earlier methods relied on a linearization of the system dynamics and measurements, which could lead to suboptimal solutions if the motion were not approximately linear between keyframes. This is a problem since for visual-inertial odometry systems several seconds may pass between keyframes. Therefore it can be difficult to approximate the motion with linear terms. The non-linear factor recovery method tries to restore a more accurate trajectory between keyframes by reducing the linearization errors that occur in the optimization process by re-integrating non-linear factors that were previously linearized during the initial optimization. This allows BASALT to capture the non-linear dynamics of the system, leading to more accurate state estimation. [18] As a result, BASALT presents state-of-the-art results on the EuRoC dataset for both visual odometry solutions with global map optimization disabled and for V-SLAM solutions once the module is enabled. [18]

VINS-Motion

In 2018, Qin et al. presented VINS-Motion [17]. It is a tightly coupled nonlinear optimization-based method by fusing pre-integrated IMU measurements and Harris corners detector for features tracked by the KLT sparse optical flow algorithm. Furthermore, they added a loop detection module and a 4-DOF pose graph optimization to enforce global consistency. The system is therefore not a pure visual odometry since it contains a backend optimization module, but they do show a reliable, complete, and versatile system that is applicable for different applications while its performance is comparable to that of state-of-the-art algorithms. They argue that the feature-based VINS estimator has reached the maturity of real-world deployment, but still see many directions for future research. One direction is that for mass deployment on a variety of consumer devices such as smartphones, the application requires online calibration of almost all sensor intrinsic and extrinsic parameters. [17]

2.3.3 Deep learning methods

In general, deep learning methods are still in their infancy compared to classical methods when considering Absolute Trajectory Error (ATE). However, deep learning methods show promising results for robust pose estimations and generalize well for unseen datasets. Following is a description of an impactful deep learning visual odometry estimation model, DeepVO, and a more recent development called ContextAVO.

DeepVO

In 2017, Wang et al. presented DeepVO [19]. They presented a novel end-to-end framework for monocular visual odometry by using deep Recurrent Convolutional Neural Networks (RCNNs). The end-to-end model infers poses from a sequence-pair of raw RGB images usually from a video file. Their architecture is depicted in Figure 2.4.

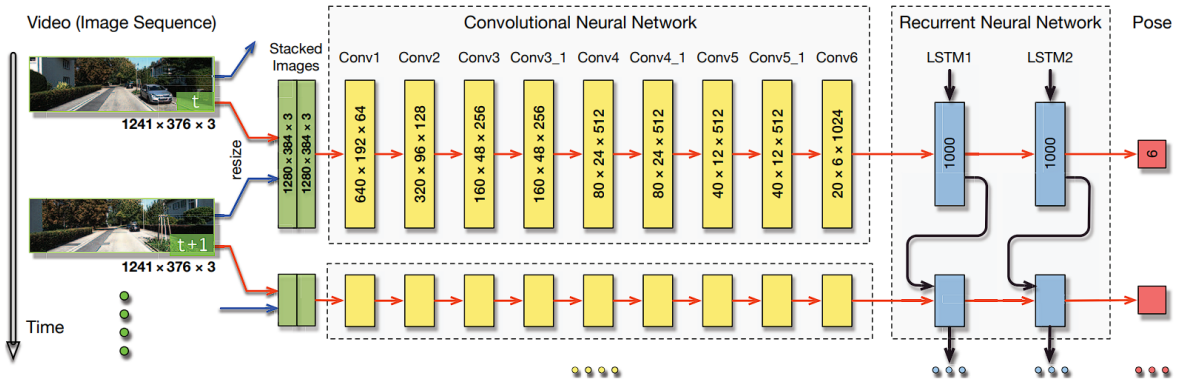


Figure 2.4: Network architecture presented in DeepVO, each pose estimate relies on two frames which is passed through a CNN layer for feature extraction then to a RNN for the pose estimation. Source: Wang et al. [19]

The approach presented by Wang et al. replaces all the modules in a classical handcrafted visual odometry pipeline with deep learning. It leveraged the capability of CNN to extract features from the images, and then a Recurrent Neural Network (RNN) to capture the temporal information among sequential frames. The end-to-end approach in DeepVO allowed the network to be optimized as a whole. This means that the modules of the networks could influence each other during learning. For instance, the features required for RNN to learn the geometric relationship between frames influenced the CNN module to focus on feature maps that were specialized for tracking, this connection is also depicted clearly in the architecture in Figure 2.4. DeepVO proved that a deep learning-based visual odometry algorithm could achieve state-of-the-art results on the KITTI visual odometry benchmark set. The model also transfers well to new scenarios. The researcher behind DeepVO stresses that incorporating classical geometric approaches with the representation, knowledge, and models learned by the Deep Neural Networks will further improve accuracy and most importantly, robustness. [19]

ContextAVO

In February 2023, Song et al. presented ContextAVO. Their approach focused on the effectiveness of local contexts to improve the estimation recovered from continuous multiple optical flow snippets. They introduce Context-Attention Refining into a novel learning-based visual odometry framework to enable the model to better capture relevant information and ignore irrelevant noise. Additionally, they introduce a multi-length window in the input. This is done by applying three sliding windows with different sizes to select continuous optical flow for an input system. This was implemented to make the system more suitable for general scenarios instead of relying on a fixed input length based on empirical knowledge. As a result, ContextAVO achieved comparable results to classical approaches and showed better performance than state-of-the-art deep learning visual odometry solutions.[20]

2.3.4 Hybrid methods

Hybrid methods attempt to harvest the strengths of traditional hand-crafted approaches with the robustness of deep learning. Many of the state-of-the-art direct and indirect methods have been enhanced by supplementing their pipeline with deep learning methods. Below is a description of such methods.

SalientDSO

In 2019, Liang et al. attempted to enhance DSO by using deep learning methods to help reduce the number of samples in the image required for pose estimation and increase the pose estimation quality. In SalientDSO [21], Direct Sparse Odometry (DSO) was extended by using a saliency map. The saliency map which is a piece of high-level semantic information about the scene is utilized in the point selection strategy in DSO. It improved DSO by making it more robust as well as more accurate. Weighting the point selection with a saliency map

required fewer points per frame to achieve convergence since the overall quality of the selected points was greater. This also resulted in a faster computation of the error term, however, the pre-processing step of generating a saliency map slowed down the overall implementation. The saliency map is made by using SalGAN, a deep convolutional neural network for visual saliency prediction trained with adversarial examples. As a result, SalientDSO obtained more accurate results as well as an increase in robustness for challenging conditions. [21]

DeblurSLAM

In 2021, Guo et al. presented "DeblurSLAM: A Novel Visual SLAM System Robust in Blurring Scene" [19]. They argue that for V-SLAM feature point extraction and tracking are closely related to image quality. Bad image quality such as severe motion blur will reduce the accuracy of the system. Motion blur in the image can appear due to fast camera motion, long exposure time due to low light environment, or a highly dynamic scene. To help the system deal with reduced image quality due to motion blur, they introduced a deblurring module to the ORB-SLAM2 consisting of a blur detection block and a deblurring block based on DeBlurGANv2 as shown in Figure 2.5. Otherwise, the pipeline depicted in the Figure shows the ORBLSAM2 procedure.

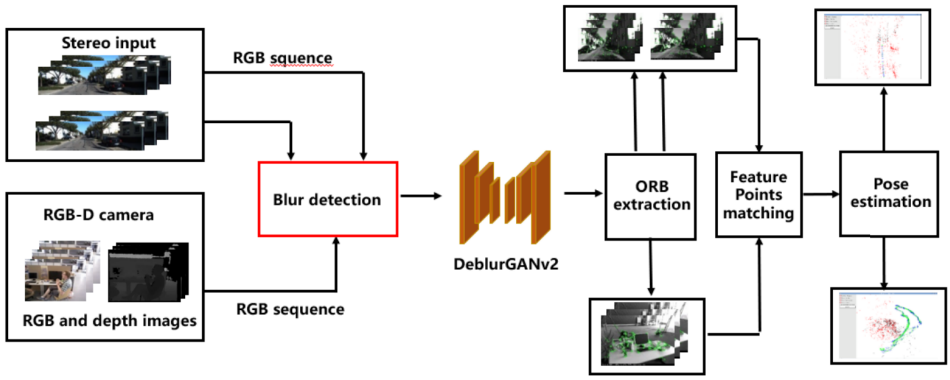


Figure 2.5: Implementation pipeline of DeblurSLAM made by Gou et al. [22]. The figure shows how blur detection and blur removal are implemented before being passed to the ORBSLAM module.

They used a GAN-based deblurring network called DeblurGANv2 because of its strong ability to handle blur and great advantage in computing speed. Their results demonstrated that DeblurSLAM outperforms the accuracy of standard V-SLAM baselines such as ORB-SLAM2 without the deblurring module. [22] The performance gap is small when evaluated on popular benchmark datasets such as the KITTI dataset, however, if a dataset contains strongly blurred scenes this deburring module would show a greater impact.

MBA-VO

In 2021, Liu et al presented Motion blur-aware visual odometry [23]. They extended DSO to consider the camera's local trajectory within the exposure time. This allowed them to compensate for motion blur that occurs due to the camera motion. They build upon the direct image alignment presented by DSO. In an extension of DSO, they parameterize two camera poses, one for the beginning of the exposure and one at the end. Then they linearly interpolate between the poses as a function of the exposure time. This way they can calculate the amount of motion blur if the two poses are far apart or close to each other. As DSO relies on photometric consistency, to continue tracking they need to either deblur or re-blur the keyframe to maintain the consistency between frames. MBA-VO chooses to re-blur the keyframe since it is computationally easier and more robust compared to motion deblurring, especially if the blur is severe. As a result, they achieve superior performance on their own virtual heavily blurred dataset generated with Unreal Engine. They choose some select sequences from the TUM RGB-D dataset and achieve better results than with DSO but fall behind ORBSLAM on the same sequence. [23]

2.3.5 Summary of related work

The different main approaches to visual odometry have been presented. Among the methods were direct, indirect, deep learning, and hybrid. Direct methods, together with an IMU have shown superior performance for visual odometry. Indirect methods are a great choice for full SLAM systems, and deep learning methods are still developing but have shown potential in modeling complex relationships in the data. Hybrid methods have shown the capability to combine the accuracy of the direct methods with the robustness of deep learning methods. Therefore, from here on the project will focus on hybrid methods. The next section will investigate the challenges for visual odometry for fields that may be alleviated by deep learning.

2.4 Challenges in visual odometry

This and the following sections will address the current challenges visual odometry is facing in its development. Factors that challenge visual odometry in producing accurate and reliable estimates are: [24] [25]

- Accurate sensor calibration
- Scale ambiguity
- feature detection and tracking
- Robustness to new environments
- Robustness to lighting conditions
- Handling dynamic environments
- Computational complexity
- Long-term operation
- Degenerate motion

Handling all of the bullet points above will create a very capable visual odometry system. However, only a selection will be discussed in more detail by their general applicability to most implementations. For instance, the requirement to extensively explore the effects of degenerate motion and computational complexity depends on the implementation. Instead, focusing on calibration, scale, feature detection, and tracking is vital for every visual odometry system, these points will be discussed next.

2.4.1 Accurate sensor calibration

Obtaining an accurate sensor calibration is crucial when modeling the environment based on the incoming light hitting the sensor. An inaccurate calibration will lead to drift and tracking errors in a visual odometry system. [25]. Unaccounted lens distortion is visualized in Figure 2.6. The figure shows the distortion effects of a wide field of view lens as well as the calibrated image when distortion is accounted for. However, even if an extensive calibration is performed prior to running a visual odometry algorithm, it might not hold in real-world applications. Rapid motion or accidental dents may cause material fatigue, and thermal expansion and contraction will deteriorate the calibration over time. Therefore a calibration cannot be assumed to be constant and the system must therefore be robust enough to handle inaccuracies in the calibration or be able to re-adjust its calibration while running. The stereo cameras from Carnegie Robotics are known for their rugged cameras that can operate for long periods without requiring a re-calibration. This is a result of extensive hardware testing and carefully construct and select materials for the design.

(a) Distorted image from the color imager. The wide lens increases the field of view but distorts the image.

(b) Rectified image using the calibration supplied by Carnegie Robotics

Figure 2.6: Image capture using Carnegie Robotics' MultiSense S30 color imager.

2.4.2 Scale ambiguity

All of the presented solutions in related work are monocular visual-inertial odometry systems. A common problem for monocular visual-inertial odometry is that scale cannot be directly inferred from the visual data. In a monocular setup, it is impossible without prior knowledge to know whether an object is large or just appears further away. This is an inherent problem, since when the 3D point in the world is projected onto the 2D image plane of a camera, depth information of that point is lost. This can lead to additional drift and incorrect scale estimates for the trajectory. If the scale is not corrected in a monocular visual-inertial odometry system, a tracking result such as depicted in Figure 2.7 can be expected.

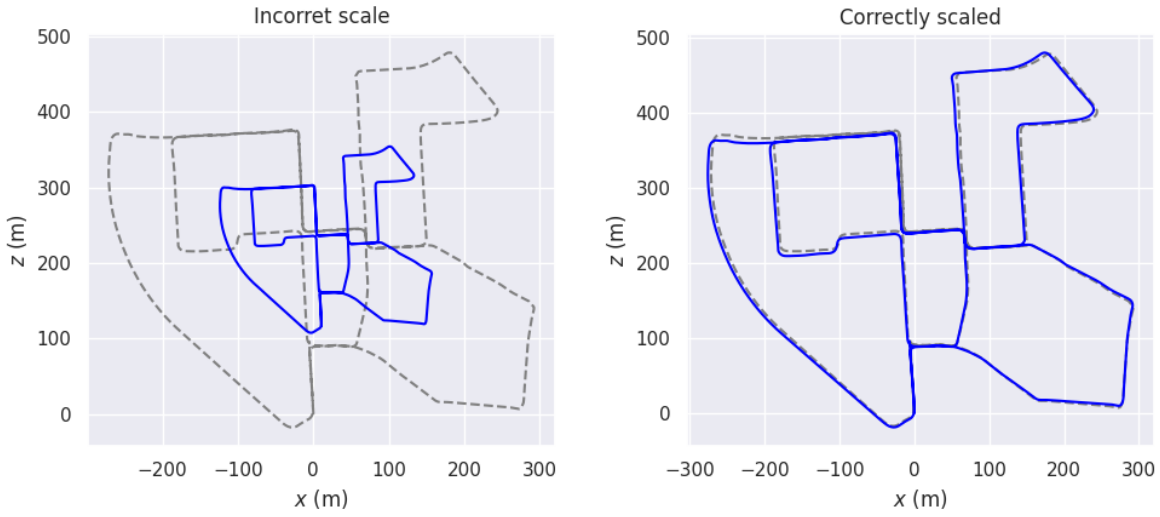


Figure 2.7: The scale ambiguity problem illustrated on a sequence of the KITTI dataset. The estimated trajectory is a result of monocular ORBSLAM which has not been initialized with the correct scale.

To solve the ambiguous scale problem, one can put markers with a known size in an initialization step to set the correct scale for the visual odometry estimation. However, this can not be extended to unknown environments. Other approaches have leveraged deep learning for depth estimation in monocular images which can be used to infer scale. By training a model on a large dataset with ground truth scale information, the model can learn to predict the scale factor for unseen environments. [25] DM-VIO uses an Inertial Measurement Unit (IMU) to attempt to optimize for the scale [14]. Other approaches have simply used another camera in a stereo camera setup to triangulate the position of landmarks. The known baseline between the cameras can be used to eliminate scale ambiguity and determine the size of the camera movement. This is also what is used in production for Carnegie Robotics where all of their cameras used for depth estimation are based on stereo vision setups.

2.4.3 Handling dynamic environments

Dynamic environments pose more challenges for a visual odometry system. The motion of objects in the scene can be misinterpreted as camera motion, leading to false motion estimates. If the majority of the features or the sampled pixels are on a moving object, the visual odometry system may incorrectly estimate the camera's motion. [26] Another aspect is occlusion, dynamic objects can occlude and obscure static features in the environment, which makes it difficult to track these features consistently. Specific examples are robotics used outdoors in offroad terrain. One may want to employ autonomous monitoring of forest ecosystems, track indicators of illegal logging, or observe and follow wildlife movements and patterns. visual odometry systems will struggle to try to cope with the dynamic lightning and movements of animals, birds, or leaves fluttering in the wind. Another example is cars driving on the road. An image from the KITTI dataset is depicted in Figure 2.8. It can be difficult to separate moving objects from static objects in the scene. Especially if a car is moving or parked. If points or features belonging to a moving object is included in the pose estimation the static scene assumption does not hold for the presented methods in the previous section, which will have a negative impact on the tracking accuracy.

Figure 2.8: Dynamic scenes of the KITTI dataset. From a single image, it is difficult to determine if the cars are moving or parked

Handling dynamic environments is a hot topic in visual odometry research and is yet to be handled. Attempts have emerged trying to use deep learning to gain a high-level perception of the scene which then can be utilized to avoid sampling from the dynamic objects. Vertens et al. attempted to use semantic motion segmentation using deep convolutional networks. They obtained an object label and the motion status of each pixel in an image. [27] An example output of their work is seen in Figure 2.9. The semantic labels can then be used to identify dynamic areas of the scene which can then be subtracted from the optical flow or ignored by the visual odometry system. However, accurately performing pose estimation in highly dynamic scenes is still a hot topic and an unsolved problem.

Figure 2.9: Motion segmented using the SMSnet on an image of the Cityscapes-Motion dataset, presented by Vertens et al. The blue overlay shows parked and non-moving cars, while the green labels show moving cars. source: Vertens et al. [27]

2.4.4 Feature detection and tracking

The classical and hand-crafted approaches usually rely on feature detection and tracking. A reliable and consistent system for detecting and tracking is essential for visual odometry. However, this can be challenging for systems operating in environments with repetitive textures or textureless areas, occlusions, or motion blur. Such environments can result in wrong correspondences between frames and further on tracking failures.

Therefore, for both indirect and direct methods, it is important to sample from image areas that offer data variation. For the indirect methods, feature detection and tracking such as ORB features achieve better performance on image patches that have unique corners that can be extracted from shifting viewpoints. For the direct case, the minimization problem using the pixel intensities is better conditioned if the sampled pixels include variation. For instance, if two consecutive tracked frames include mostly a white wall, most of the data variation will originate from noise in the imager which will degrade the tracking result. Another word for data variation in an image texture. An image with a lot of texture will also include data variation and to detect the amount of texture, there are a few different methods.

Texture detection

A simple and computationally efficient method for measuring the variation in pixel intensities is the local standard deviation. It is often used for pre-processing images before tasks such as feature extraction or tracking. Local standard deviation can also be used to help edge detection. Another method is the Histogram of Gradients (HOG). HOG captures the distribution of edge orientations in an image and is particularly effective at detecting object boundaries and corners. It is a more complex but powerful texture descriptor but it is not as quick to either run or

implement as the local standard deviation. An example of HOG and local standard deviation operations on a test image is shown below in Figure 2.10.

(a) Original image from the EuRoC MAV dataset.

(b) Local standard deviation applied to the original image, highlighting areas of high pixel intensity variation.

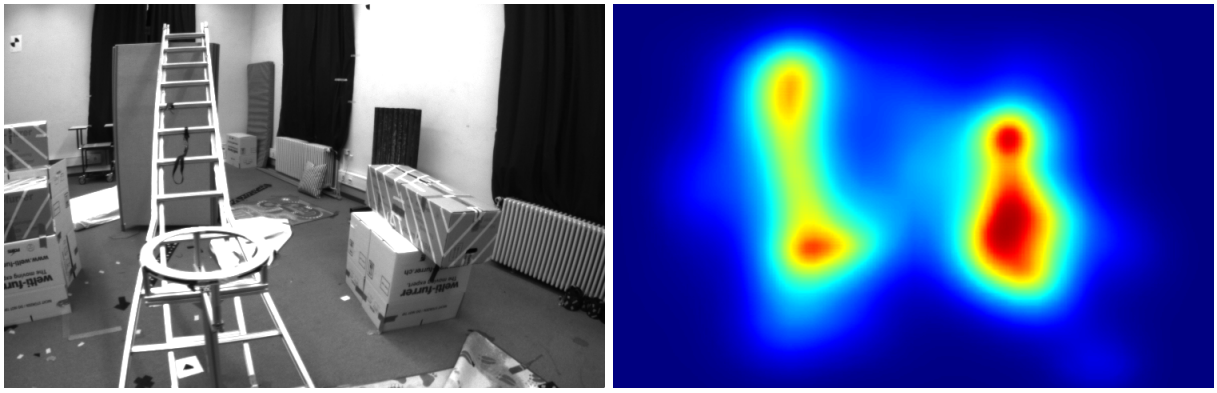
(c) Histogram of Gradients applied to the original image, capturing the distribution of edge orientations.

Figure 2.10: Illustration of texture detection methods applied to an image from the EuRoC MAV dataset. From left to right: original image, local standard deviation, and Histogram of Gradients.

However, it has demonstrated that image gradients are useful for data extraction in an image. For instance, the Direct Sparse Odometry (DSO) implementation uses image gradients to sample points, and the result is sharing some similarities to HOG. In 2021, Zeng et al. presented "Robust Mono visual-inertial odometry Using Sparse Optical Flow With Edge Detection" where the edge detector was based on the laplace operator to identify texture rich areas to sample points for the Lukas-Kanada optical flow algorithm. [28] However, as can be seen in the example with HOG descriptors in Figure 2.10c, a lot of information is lost. The loss of information is apparent by observing the smaller details in the image such as the texture on the cardboard box or at the wires lying on the floor. SalientDSO extended DSO's implementation to not only look at gradients but also saliency maps inspired by the way humans process visual information. They gathered a pixel-wise map that could be used to identify texture-rich areas.

Saliency maps

Visual attention is an important mechanism that allows humans to select the most relevant information from a visual scene. A visual saliency map is defined as the amount of attention each pixel receives by a human observer. The result is usually a heatmap where brighter pixels indicate a higher saliency value. An example of using the same picture in Figure 2.10a with saliency detection overlaid with the jet color map is seen below in Figure 3.3b. From the Figure, it can be observed that the saliency map is good at detecting texture-rich areas which the gradient approach in Figure 2.10c did not.



(a) Original image from the EuRoC MAV dataset. **(b)** Visual saliency map applied to the original image.

Figure 2.11: Illustration of visual saliency detection applied to an image from the EuRoC MAV dataset. The saliency image is overlaid with the Jet colormap to highlight regions of high saliency.

Accurately predicting saliency as perceived by humans is still an academic challenge. Past approaches such as used by SalientDSO use deep convolutional networks to extract features and predict their saliency. [21] However, a limitation of CNN is that it receives information from only a local subset of pixels which omits long-range contextual information. When humans perceive an image, the foveal vision provides high-resolution information while peripheral vision provides low-resolution but long-range information. This property can be beneficial for predicting visual saliency closer to ground truth. Some approaches have added long-range modeling capabilities by using Long-Short Term Memory (LSTM)-based architectures. This approach refined the long-range visual information and improved the visual saliency result. However, more work is needed to close the gap between saliency prediction and human perception. [29] However, SalientDSO demonstrated using a saliency predictor to weigh in on point sampling has a positive effect on tracking accuracy.

2.5 Summary and final problem formulation

The initial problem formulation was stated as such:

"What is the current state of the V-SLAM field and what obstacles does it encounter in real-world applications?"

This chapter has given a description of the V-SLAM technology. The project's scope was narrowed down to focus on the front end of a V-SLAM system, and more concisely the visual odometry part. Since visual odometry is a fundamental part of V-SLAM, the findings can still be applied to the V-SLAM system as a whole. Continuing this a detailed analysis of state-of-the-art methods for visual odometry was given within direct, indirect, and deep learning methods. This chapter ends with a description and examples of challenges in visual odometry. From this description, a problem and approach will be selected to guide the implementation of

this project. Considering the high accuracy of the direct visual-inertial odometry systems and the robustness introduced by deep learning systems, this project will focus on hybrid methods in an attempt to combine the advantages of traditional and deep learning approaches, and more specifically the traditional DSO by Engel et al. [15] and the hybrid method, SalientDSO by Liang et al. [21]. DeBlurSLAM and MBA-VO presented in related work achieved better performance in their respective implementations when considering motion blur. VI-DSO added an inertial measurement unit to minimize the error induced by rapid movements, pure rotations, and bad image quality such as low illumination and motion blur. This project will contribute to investigating how much hybrid approaches can enhance visual odometry systems. The main focus will be to rely on extensive pre-processing of data in visual odometry inspired by the approaches in SalientDSO and DeBlurSLAM. The pre-processing will also focus on factors that the IMU was targeted to solve on VI-DSO. Bad image quality as a result of motion blur was handled by an IMU, however, it is interesting to see how adding a deblurring model to DSO compares to the integration of the IMU.
The final problem formulation for this project is:

How can the integration of deep learning enhancements augment direct visual odometry to better handle rapid camera movements and adverse imaging conditions?

The final problem statement defined above will guide the implementation of the project. However, to measure the success a progressive list of success criteria will be defined.

2.5.1 Success criteria

The success criteria will guide the implementation of the project. They will be defined in a progressive order and categorized into three phases. The first phase is an initial system development phase which concerns having a working system that achieves the intended functionality. The second phase is the testing and evaluation of the system. Here the system will be compared to other pre-existing successful approaches. The last phase consists of an elaborate and extensive analysis of edge cases to understand the limitations and points of failures of the system. The list of success criteria is listed in below in Table 2.2.

Criterion No.	Description
1	Develop a framework that integrates the modules from SalientDSO, DeBlurSLAM, and DSO. This involves coding and debugging to ensure the proper functionality of the integrated system.
2	Validate the functionality of the system with simple test cases, ensuring a working implementation.
3	Evaluate the hybrid system performance against a well-recognized public benchmark dataset.
4	Achieve improved tracking accuracy in comparison to baseline methods: DSO, SalientDSO, DM-VIO.
5	Compare the tracking accuracy of the proposed method with a deblurring method to a direct Visual-Inertial odometry method.
6	Assess the computational efficiency of the hybrid system. The hybrid system should operate within an acceptable computational time and resource usage.
7	Lastly, analyze failure cases of the developed framework to understand the limitations of the hybrid system. This can provide valuable insights for future improvement.

Table 2.2: Success criteria for the project implementation

However, it is worth mentioning that the handling dynamic environments challenge described in the previous section is also a worthwhile research direction for the project. But it is not further investigated in favor of exploring hybrid direct visual odometry.

To the extent of available knowledge, no other project has used a combination of deblurring and saliency prediction to help point sampling in a direct visual odometry pipeline. The following chapter will elaborate on the implementation process devised to fulfill both the final problem formulation and the success criteria list.

Chapter 3

Implementation

This section will describe the implementation of this project and in-depth explanation of the previously used work. A high-level implementation pipeline is seen in Figure 3.1. The Figure shows how deep learning networks is used in the preprocessing step of the data as well as using the deblurred image in the point sampling strategy. Firstly a description of the deep learning models used for data pre-processing will be supplied. Secondly the Direct Sparse Odometry [15] and SalientDSO [21] will be explained. Lastly in this chapter, the implementation of the point sampling strategy will be given before being handed to DSO's pose estimation pipeline.

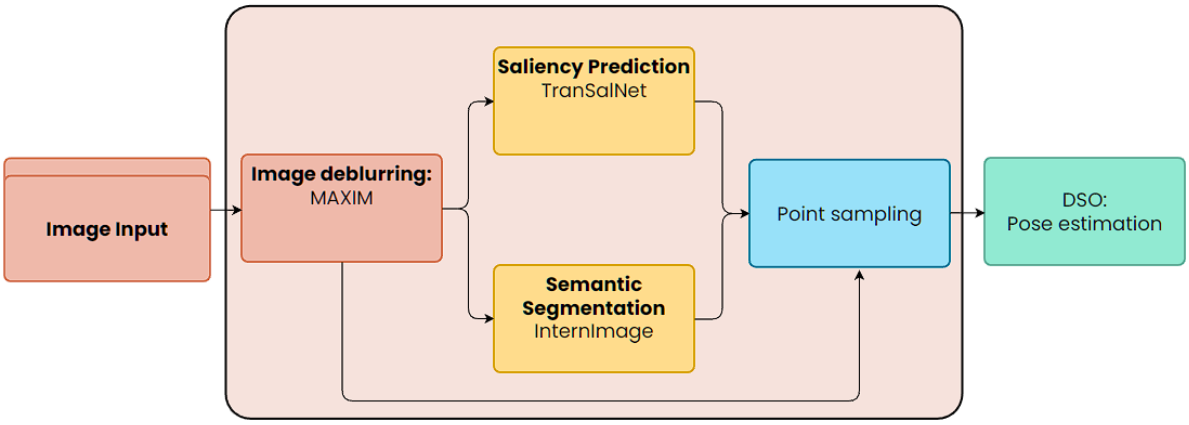


Figure 3.1: Implementation pipeline for the proposed system. The first module in the system is a Deblurring network, which preprocesses the data before being passed onto a saliency predictor and a semantic image segmentation model. The output images from deblurring, saliency, and segmentation are passed to a point sampler which feeds the direct sparse odometry formulation with input data.

3.1 Preprocessing with deep learning networks

The first step in the implementation pipeline depicted in Figure 3.1 is the pre-processing of the input data. There are three different kinds of pre-processing networks. Firstly the input data is run through a deblurring network. This reduces the image noise induced by camera movement. This part is inspired by DeBlurSLAM. Following this, an implementation similar to SalientDSO is used by using the deblurred images for semantic segmentation and saliency prediction to identify the texture-rich areas in the image. Since SalientDSO's release in 2019, models have achieved better performance at the aforementioned tasks, therefore newer and improved models are used. All of the models used are pre-trained standard models without fine-tuning supplied by the original authors of each model. To achieve the best results, each model selected is a state-of-the-art performer in its respective category and open-sourced. DSO is primarily made for real-time performance of visual odometry pose estimation and in a real-world example, the networks used would benefit from being specialized and lightweight networks. However, with a proof of concept approach, the models used have not focused on that part but lean more towards accuracy. The deblurring module may spend upwards of seconds just to process one image, the saliency predictor and segmentation network takes somewhere around 0.5 seconds to run individually. The models selected are provided by leading entities in the deep learning field such as Google Research, which provides the deblurring module, and OpenGVLAB, which supplies the semantic segmentation module. Next will be a description and an example for each of the pre-processing modules: Deblurring, Saliency predictor, and image segmentation.

3.1.1 Maxim: Deblurring

MAXIM: is a multi-axis multi-layer perceptron-based architecture that serves as a general-purpose vision backbone for image processing tasks. It was published in April 2022 and shows significant advances over the state-of-the-art in five image processing tasks in terms of peak signal-to-noise ratio. The five areas are Denoising, Deblurring, Deraining, Dehazing, and Enhancement. [30] Large-scale vision models have obtained great success on many high-level applications, but these models for low-level enhancement such as the aforementioned metrics have not been as thoroughly studied. Pioneering works on transformers for low-level vision only accepted small patches of fixed sizes, due to the intense computational requirements for self-attention. This has a bad effect and can cause boundary artifacts when cropping larger images. Local-attention-based transformers ameliorate this issue but are still limited in the receptive field, or lose non-locality. A key design in MAXIM is the use of multi-axis approach that captures both local and global interactions in parallel. By mixing the information the MLP-based operator becomes convolutional and scales linearly with image size, which is great for performance and flexibility. [30] Below in Figure 3.2 is an example of the model performing deblurring on one of the images in a visual-inertial odometry dataset. The model used was a pre-trained model given by the authors. This model is particularly effective for blurred scenes with fast camera movement or long exposure times due to low illumination. In the Figure,

it is clear that the mattress and the table on wheels in the right of the image have received enhancement and especially the wires lying on the ground in the left part of the image.

(a) Original image from the V02_03_Difficult sequence in the EuRoC MAV dataset. [**Burri25012016**]

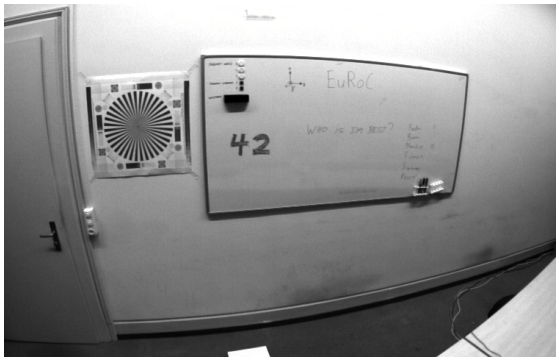
(b) Deblurred image of Figure 3.2a using the pre-trained MAXIM model.

Figure 3.2: Comparison of original dataset image and deblurred image

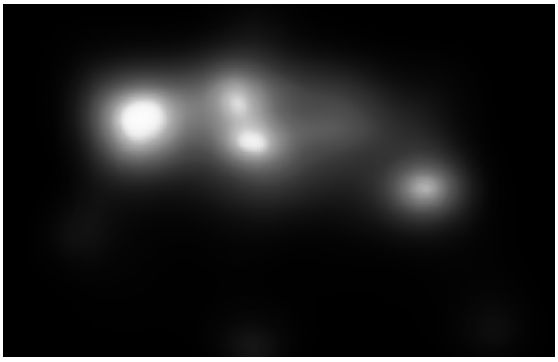
The output from the MAXIM model is used as the input for the next step which is the saliency predictor and image segmentation networks.

3.1.2 TranSalNet: Saliency prediction

TranSalNet was published by Lou. et al. in October 2021. It advanced towards a perceptually more relevant saliency prediction using deep learning models with transformers. Previous attempts at saliency prediction were largely based on CNN architecture. However, due to the inherent inductive bias of CNN encoder architectures, the extracted feature maps lack long-range contextual information, compared to the human vision system which is proficient at capturing both local and long-range visual information. Some methods combined CNN with long Short-Term Memory based components to better simulate the properties of the human attention mechanism. TranSalNet took the CNN-based models a step further and combined them with the transformer which has been successful in natural language processing (NLP), partly because of its powerful long-range dependency modeling capabilities. results showed that using TranSalNet achieved superior performance on the public benchmarks and competitions for saliency models. [29] Below in Figure 3.3 it is shown how TranSalNet predicts the salient areas of an input image for a visual odometry Benchmark dataset. The hotspots in Figure 3.3b correspond well with the textured areas in Figure 3.3a. A lot of attention is given to the poster between the door and the whiteboard, and to the text written on the whiteboard.



(a) Deblurred image from the EuRoC MAV dataset using the MAXIM model.

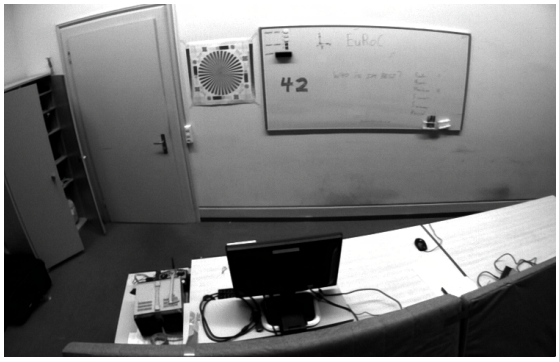


(b) Saliency predicted image of Figure 3.3a using the pre-trained TranSalNet model.

Figure 3.3: Image illustrating the saliency predictor TranSalNet on the EuRoC MAV dataset.

3.1.3 InternImage: Image segmentation

INTERN-2.5 is a powerful multimodal multitask general model based on InternImage foundation model. It was released in March 2023 by Wang et al. Its main approach was to explore large-scale models based on CNNs such as the large-scale vision transformers (ViTs) has been in recent years. InternImage stands out from other models by taking deformable convolution as the core operator. By deforming, InternImage can utilize the advantage of a large receptive field for tasks such as detection and segmentation. Furthermore, the deformable convolutions also have the adaptive spatial aggregation condition by input and task information. As a result, InternImage reduces the inductive bias inherent to traditional CNNs, this is shown as increased performance on challenging image datasets such as ImageNet, COCO, and ADE20K, and achieved a new record on COCO and ADE20K outperforming current leading CNNs and ViTs.[31]
Below in Figure 3.4 is an image from the visual inertial dataset using a pre-trained model supplied by the authors of INTERN-2.5 for the segmentation task. As depicted in Figure 3.4b, the model is capable of segmenting various elements such as the floor, door, wall, closet, desktop, and even some smaller objects. However, InternImage was primarily trained on RGB images, whereas the dataset in use consists exclusively of grayscale images. The network performance could potentially be improved by fine-tuning it for grayscale images or retraining it on grayscale datasets.



(a) Deblurred image from the EuRoC MAV dataset using the MAXIM model.

(b) Segmented image of Figure 3.4a using the pre-trained INTERN-2.5 model with the ADE20K color palette

Figure 3.4: Comparison of original dataset image and deblurred image

The next step in the implementation pipeline depicted in Figure 3.1 is the point sampling. However, to understand the importance of the method by which points are sampled, an overview of the direct image alignment formulation used in DSO will be explained, furthermore, an explanation of the changes made by SalientDSO will be introduced.

3.2 Direct visual odometry implementation

In 2016 J Engel et. al. presented Direct Sparse Odometry (DSO), which is based on a novel, highly accurate sparse and direct structure and motion formulation. It is a probabilistic model with consistent joint optimization of all model parameters. This includes geometry represented as inverse depth in a reference frame, camera intrinsics, photometric parameters, and camera motion. DSO assumes photometric consistency between each image. This means that the transformations between two frames can be found directly from pixel intensities in the image [23]. To rely on this consistency an accurate formulation of the image-forming process is required. The next section will dissect the calibration process used in DSO.

3.2.1 Camera calibration

The calibration step in DSO consists of two different calibrations. Firstly a geometric calibration is responsible for camera intrinsics such as focal length, principal point, and lens distortion, and secondly, a photometric calibration accounts for the intensity changes in statically lit scenes.

Geometric calibration

DSO's geometric calibration is made for the pinhole camera model, and radial distortion caused by the lens is removed in a pre-processing step. The pinhole camera model mapping function of a point in 3D space to the 2D image plane is denoted by $\Pi_c : \Re^3 \rightarrow \Omega$ and back-projection

with a point in the image plane and a depth value $\Pi_c^{-1} : \Omega \times \Re \rightarrow \Re^3$ where $\mathbf{c}$ denotes the intrinsic camera parameters. The distortion model used for this project is the radial-tangential model which corresponds to the distortion model used by the dataset presented in Chapter 4. The radial-tangential distortion model is a combination of two expressions. The mathematical model for radial distortion is expressed as a series expansion, so given an undistorted point (x, y) in the image plane, the distorted point (x', y') is given by:

$$x' = x(1 + k_1 r^2 + k_2 r^4)$$

$$y' = y(1 + k_1 r^2 + k_2 r^4)$$

where $r^2 = x^2 + y^2$ is the squared distance from the point to the optical axis and k_1, k_2 are the radial distortion coefficients. Then, the model for tangential distortion is added to the radial distortion model. The distorted point (x', y') is then given by:

$$x' = x + [2p_1 xy + p_2(r^2 + 2x^2]$$

$$y' = y + [p_1(r^2 + 2y^2) + 2p_2 xy]$$

where p_1, p_2 are the tangential distortion coefficients. Undistorting the image is typically done as a pre-processing step before the pinhole camera model is applied. [32]

Photometric calibration:

The details of the photometric calibration is described by J. Engel et al. in [33]. From here, they calibrate the Camera Response Function (CRF) as well as pixel-wise attenuation factors. The camera response relates the scene's radiance to image brightness. A well-calibrated camera response function will more accurately relate scene radiance to image brightness. [34]. The CRF and attenuation factors G and V, is model is given by:

$$I(\mathbf{x}) = G(tV(\mathbf{x})B(\mathbf{x})) \tag{3.1}$$

Where t is exposure, B is the irradiance image and I is the observed pixel value. Note that G, V, and B are only observable up to a scalar factor. Derivation of the unknown terms $G(\cdot), V(\cdot)$ and $B(\cdot)$ can be found in [33].
To photometrically correct each video frame the following equation can then be used, where I_i' is the corrected image:

$$I_i'(\mathbf{x}) := t_i B_i(\mathbf{x}) = \frac{G^{-1}(I_i(\mathbf{x}))}{V(\mathbf{x})} \tag{3.2}$$

In subsequent equations, I_i will always refer to the corrected image I_i'. For DSO the photometric calibration is an important step for the system. This is because the model explained in the next section solely relies on the pixel intensities from the irradiance image. Therefore removing photometric noise in the image induced by factors such as the lens is a crucial step. An example of the effect the lens has on an image and the resulting correction step can be seen below in Figure 3.5.

Figure 3.5: CRF and Vignette calibration for the Carnegie Robotics' MultiSense S30 camera. The Left is the original image from the camera, and the right shows the calibrated image with response calibration and vignette map. As seen in the right image the calibration, especially around the corners the lens vignetting effect is reduced and the center is more uniformly exposed.

3.2.2 Model formulation

DSO define the photometric error of a point $\mathbf{p} \in \Omega_i$ in a reference frame I_i, observed in a target frame I_j as the weigthed sum of squared differences (SSD) over a neighborhood of pixels.

$$E_{\mathbf{p}j} := \sum_{\mathbf{p} \in N_{\mathbf{p}}} \omega_{\mathbf{p}} ||(I_j[\mathbf{p}'] - b_j) \frac{t_j e^{a_j}}{t_i e^{a_i}} (I_i[\mathbf{p}] - b_i||_\gamma \tag{3.3}$$

Where $N_{\mathbf{p}}$ is the set of pixels in the SSD, t_i, t_j the exposure times of the images I_i, I_j and $|| \cdot ||_\gamma$ the Huber norm. The variables b_j, b_i, a_j, a_i are included in an affine brightness transfer function for the images. $\mathbf{p}'$ is the projected point position of $\mathbf{p}$ with inverse depth $d_{\mathbf{p}}$ given by:

$$\mathbf{p}' = \Pi_c(\mathbf{R}\Pi_{\mathbf{c}}^{-1}(\mathbf{p}, d_{\mathbf{p}}) + \mathbf{t} \tag{3.4}$$

with

$$\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} := \mathbf{T}_j \mathbf{T}_i^{-1} \tag{3.5}$$

In addition to using robust Huber penalties, a gradient-dependent weighting $w_{\mathbf{p}}$. It is responsible for down-weighting pixels with high gradients and is given as:

$$w_{\mathbf{p}} := \frac{c^2}{c^2 + ||\nabla I_i(\mathbf{p})||_2^2} \tag{3.6}$$

The full photometric error over all frames and points is then given by:

$$E_{photo} := \sum_{i \in F} \sum_{\mathbf{p} \in P_i} \sum_{j \in OBS(\mathbf{p}} E_{\mathbf{p}j} \tag{3.7}$$

where i runs over all frames $F, \mathbf{p}$ over all point P_i in frame i, and j over all frames $obs(\mathbf{p})$ in which point $\mathbf{p}$ is visible. To summarize the DSO model formulation depends on the following variables:

- The point's inverse depth $d_{\mathbf{p}}$
- The camera intrinsics $\mathbf{c}$
- Poses of the target and tracked frames: $\mathbf{T}_i, \mathbf{T}_j$ and brightness transfer function parameters: a_i, b_i and a_j, b_j

The photometric error in equation 3.7 is then optimized using the Gauss-Newton algorithm in a sliding window approach. Which follows an approach presented by Leutenegger et al. in [35] A more detailed explanation of this step can be seen in the original paper "Direct Sparse Odometry" by Engel et al. [15] and Leutenegger et al. [35].

3.2.3 Point selection strategy

The point selection strategy follows a different approach than other direct methods. Engel et al. found that image data is highly redundant, and the benefit of using more data points quickly flattens off. They aim to keep a fixed number N_p of active points, $N_p = 2000$, equally distributed across space and active keyframes, in the optimization. In three steps the point sampling and point management are as follows: [15]

1. Candidate points are tracked individually in subsequent frames, generating a coarse depth value that will serve as initialization for the optimization. The requirement for candidate points aims at selecting points that are well-distributed in the image and have sufficiently high image gradient magnitude with respect to their immediate surroundings. They obtain a region-adaptive gradient threshold by splitting the image into 32x32 blocks and for each block, they compute a gradient threshold over all the pixels in that block. They also found that it was often beneficial to also include some points with weaker gradients when no high-gradient points are present. To achieve this, the candidate point selection procedure was repeated twice more width decreased gradient threshold and increased block size to generate the desired amount of points,

2. Point candidates are tracked in subsequent frames using a discrete search along the epipolar line, minimizing the photometric error 3.3. They compute a depth and associated variance, which is used to constrain the search interval for the subsequent frame. This tracking is inspired by LSD-SLAM. The computed depth only serves as initialization once the point is activated.

3. After a set of old points is marginalized, new point candidates are activated to replace them. They aim to maintain a uniform spatial distribution across the image. To do this, they project all active points onto the most recent keyframe, then activate candidate points that maximize the distance to any existing point all within the keyframe.

An example of DSO's point sampling from a scene in the EuRoC MAV dataset in seen below in Figure 3.6

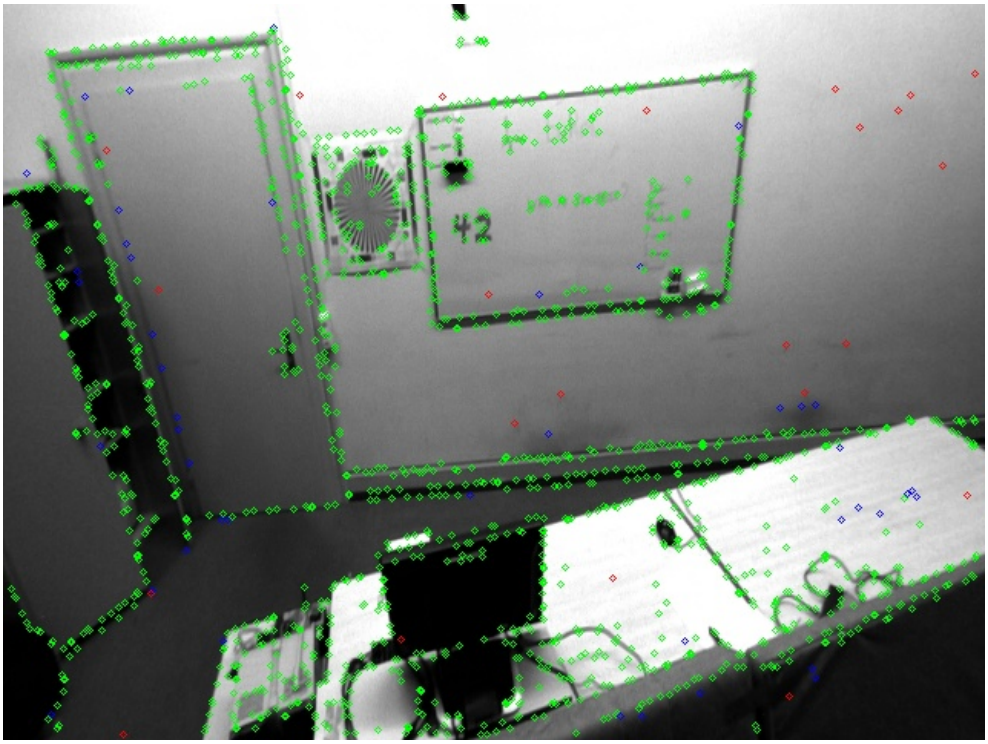


Figure 3.6: Points selected from a difficult sequence from the EuRoC MAV dataset using the baseline DSO method

3.3 SalientDSO implementation

SalientDSO's framework is composed of a preprocessing step and a Visual Odometer backbone. The visual odometry backbone is adopted from DSO. The preprocessing step involves saliency prediction and scene parsing using deep convolutional neural networks and using the outputs to help the point selection strategy. [21] This section will describe in more detail the visual saliency prediction, and filtering saliency using semantic information and how it affects the point sampling in DSO.

3.3.1 Visual saliency prediction

Saliency prediction in general is a difficult problem because it envelopes how humans process visual information. Lately, data-driven approaches have excelled at this task. SalientDSO adopted SalGAN for saliency prediction. In brief, SalGAN introduced the use of a Generative Adversarial Network (GAN) for saliency prediction.[21] The researchers behind SalientDSO found that the salinecy produced by SalGAN is concentrated around a fixation point inside an object and is fuzzy. Additionally, the predicted saliency map is not robust to viewpoint and illumination, meaning that when the camera moves around the fixation point does not remain constant. To solve this, they introduced Pyramid Scene Parsing (PSPNet) for pixel-

level scene segmentation prediction. Each object segmented in the scene is assigned a saliency value based on the output from SalGAN, this helps maintain focus on interesting objects in the scene. However, some empirical knowledge was introduced. They manually down-weighted the saliency of uninformative regions such as walls, ceilings, and floors. The semantic filtered saliency map $\hat{S}_j^{weighted}$ was given by:

$$\hat{S}_j^{weighted} = w_C(C_j)\hat{S}_j \tag{3.8}$$

Where w_C are predefined weights obtained empirically for different classes. To create a smoother saliency map they also replaced each pixel by the median of saliency for its respective class such that:

$$\hat{S}_j^{final} = median\{\hat{S}_j^{weighted}, \forall i \in C_j\} \tag{3.9}$$

A final semantic filtered saliency image is seen in Figure 3.7. Note that the image segmentation was done with InternImage and saliency prediction was done with TranSalNet due to being unable to reuse SalGAN and PSPNet on the hardware for this project. This is expected to have some impact on SalientDSO as a baseline method for comparison, however, it will presumably favor it positively by having more accurate scene segmentation and saliency prediction.

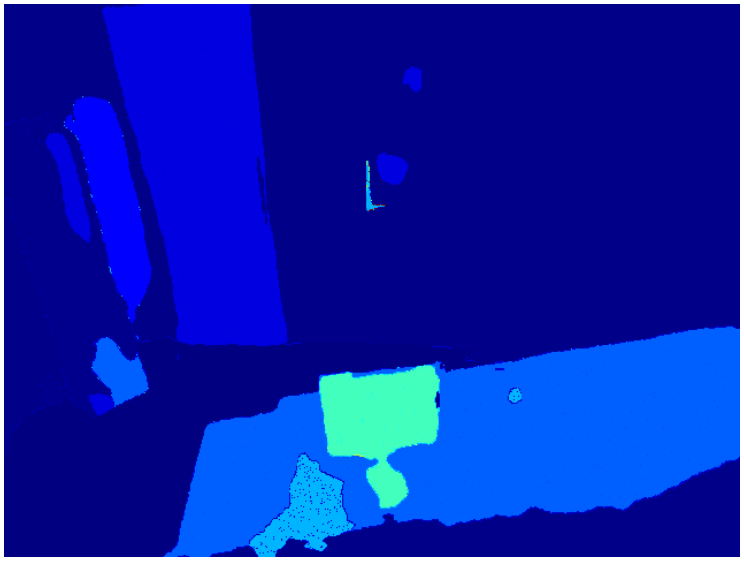


Figure 3.7: Saliency and segmentation filtered version of the image of Figure 3.4

The next section will explain how the segmented saliency image affected the point selection strategy.

3.3.2 Point selection strategy

While DSO samples candidate points that were uniformly spatially distributed, SalientDSO selects points based on saliency. To do this they split the image into KxK patches. For a patch

M_i they compute the median of gradient as a region-adaptive threshold, and the median of saliency as a region-adaptive sampling weight sw_i. For each patch, the sampling weight is computed as:

$$sw_i = median\{\hat{S}_j^{final}, \forall j \in M_i\} + s_{smooth} \tag{3.10}$$

Where s_{smooth} is a laplacian smoothing controlling the bias on a salient region. The probability of a patch M_i from all patches M being sampled is:

$$\mathbf{P}_S(M_i) = \frac{sw_i}{\sum_{m \in M} sw_m} \tag{3.11}$$

A higher saliency weight will result in a higher probability of that patch being sampled. Once a patch M_i is selected the patch is then split into dxd blocks. From here the approach is the same as in DSO, where for each dxd block the pixel with the highest gradient is selected if it surpasses the region-adaptive threshold. Repeating DSO's procedure within a patch yields a good result in which the points within a patch are evenly distributed. The point selection with SalientDSO is shown in Figure 3.8.

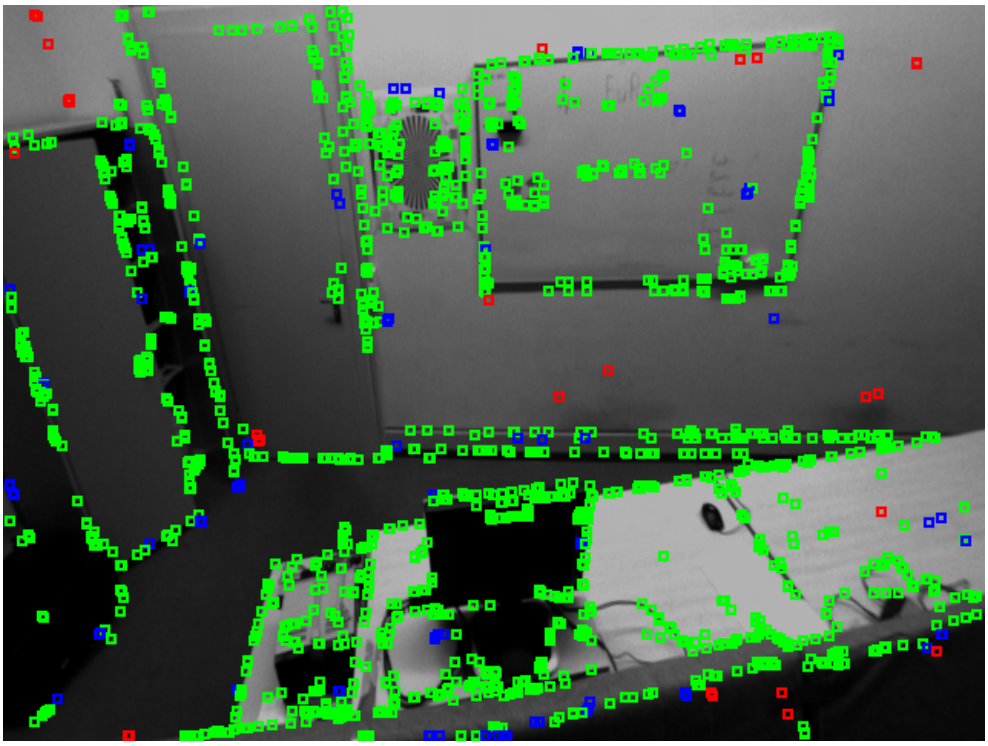


Figure 3.8: Points selected from a difficult sequence from the EuRoC MAV dataset using SalientDSO method

The modified point selection strategy is very clear when comparing the differences in Figure 3.6 and Figure 3.8. Uninformative regions have been down-prioritized and points are more strongly sampled from places with more texture variation.

3.4 Deblurred Salient DSO implementation

The Deblurred Salient DSO implementation first presented in this project builds upon SalientDSO, DSO, and DeBlurSLAM. As shown in the implementation pipeline in Figure 3.1, the data is preprocessed with an image deblurring network, a saliency predictor network, and finally an image segmentation network. The main idea behind Deblurred Salient DSO is that camera or scene movement during the camera exposure period produces noise in the resulting image that makes it difficult to restore the pose. For a method such as DSO which relies on a photometrically accurate image, this will have a noticeable effect on the tracking accuracy. The use of the MAXIM deblurring model serves two purposes. Firstly, it deblurs the input images to both the DSO and the SalientDSO predictor and segmentation networks. Secondly, the model is used to identify the blur-induced areas of the image. This allows the implementation to account for the blur. The implementation computes a pixel-wise weight map to be used together with the saliency weighting of the pixels. The pixel-wise weight map B_{diff} computed from the original image $I_{orig}(\mathbf{x})$ and the deblurred image $I_{deblur}(\mathbf{x})$, where i_{max} is normalization factor and defined as the max pixel intensity value in either image:

$$B_{diff}(\mathbf{x}) = \frac{|I_{orig}(\mathbf{x}) - I_{deblur}(\mathbf{x})|}{i_{max}} \tag{3.12}$$

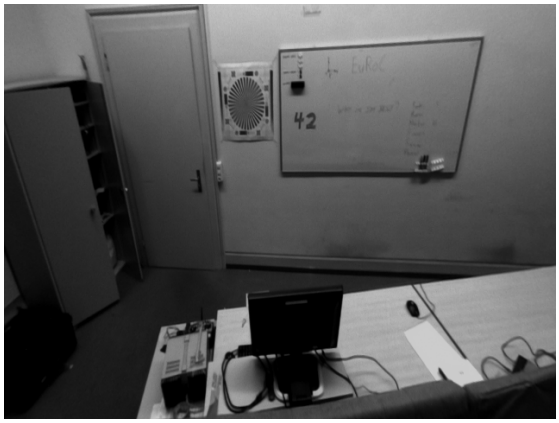
An example calculation of a scaled B_{diff} is seen in Figure 3.9b. white area i.e. the value 255 represents areas with no blur and the black areas are the regions that have been modified by the MAXIM model.

3.4.1 Changes to point selection

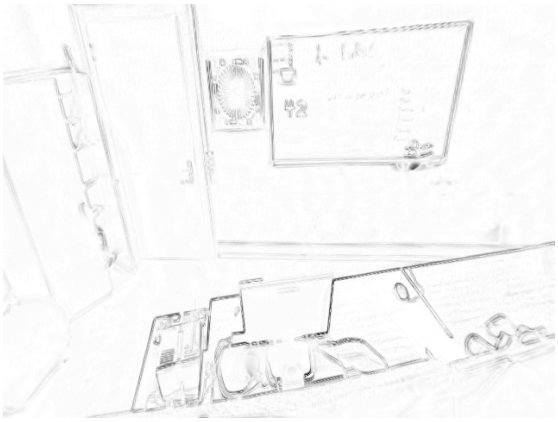
Recall the weighted saliency calculation using weighted by the segmentation map in Equation 3.8:

$$\hat{S}_j^{weighted} = w_C(C_j)\hat{S}_j$$

To calculate the final weight, two methods are attempted. Firstly the point sampler will try to avoid the deblurred areas while in the other it will favor deblurred areas of the image.



(a) Deblurred image from the EuRoC MAV dataset using the MAXIM model.



(b) Blur difference image calculated from the original in the EuRoC MAV dataset, and the deblurred image shown to the left.

Figure 3.9: Difference of pixel intensities in the original dataset image and the deblurred image

Avoid deblurred areas of the image

The idea of avoiding the blurred areas identified by the MAXIM model is based on Engel et al. argument that image data is highly redundant. This suggests that we can safely ignore those areas that have been impacted by blurring noise. To do this, the probability of a patch being sampled is affected according to the pixel value at B_{diff} image.
This results in the following equation:

$$\hat{S}_j^{blur\_weighted} = w_C(C_j)\hat{S}_j B_{diff}^2 \tag{3.13}$$

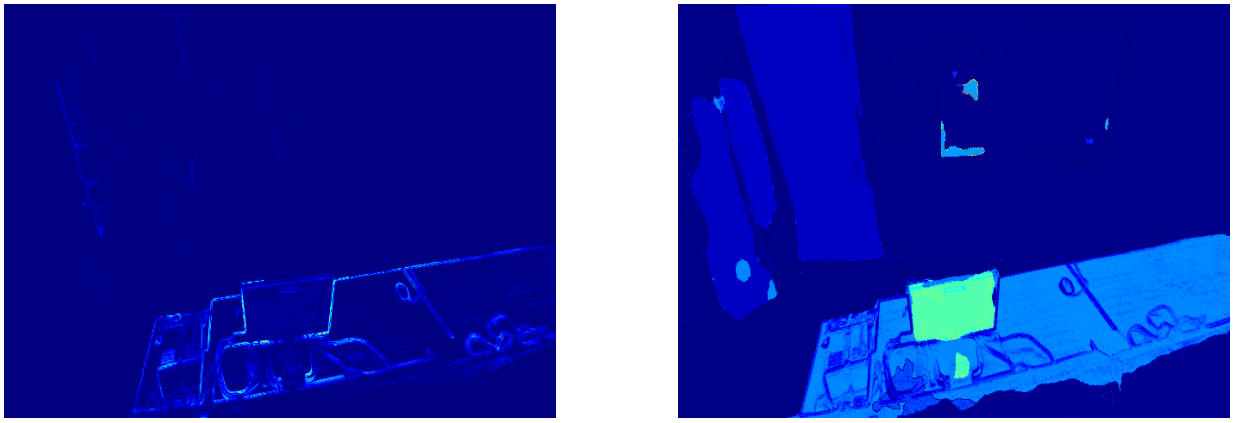
where B_{diff} is a normalized variable $[0, 1]$ where a 1 indicates no deblurring in that particular pixel compared to 0 where the deblurring model has altered the pixel. The variable has been raised to the power of two which achieved better results in preliminary testing. $\hat{S}_j^{blur\_weighted}$ replaces $\hat{S}_j^{weighted}$ in subsequent equations. An example of how this affects the final saliency weights are shown in Figure 3.11b. As shown in the image, the borders around objects are especially affected by the avoid scheme. A darker value corresponds to a lower probability of pixels being sampled from this region.

Attract the deblurred areas

If the deblurring network enhances the image's precision it may be beneficial to focus more on the areas that have undergone deblurring. Since DSO relies on the photometric consistency assumption and MAXIM modifies the image aiming for the highest signal-to-noise ratio. The deblurred areas may contain more photometrically accurate information. Therefore we want to favor the areas that have been deblurred. This results in the following equation:

$$\hat{S}_j^{blur\_weighted} = w_C(C_j)\hat{S}_j(1 - B_{diff}) \tag{3.14}$$

where B_{diff} is a normalized variable same as before and $\hat{S}_j^{blur\_weighted}$ replaces $\hat{S}_j^{weighted}$ in subsequent equations. An example of how this affects the final saliency weights are shown in Figure 3.11a. In the illustration, small pixel-wise regions containing deblurred data are highly weighted. This ensures the point sampling will focus on patches containing deblurred regions.

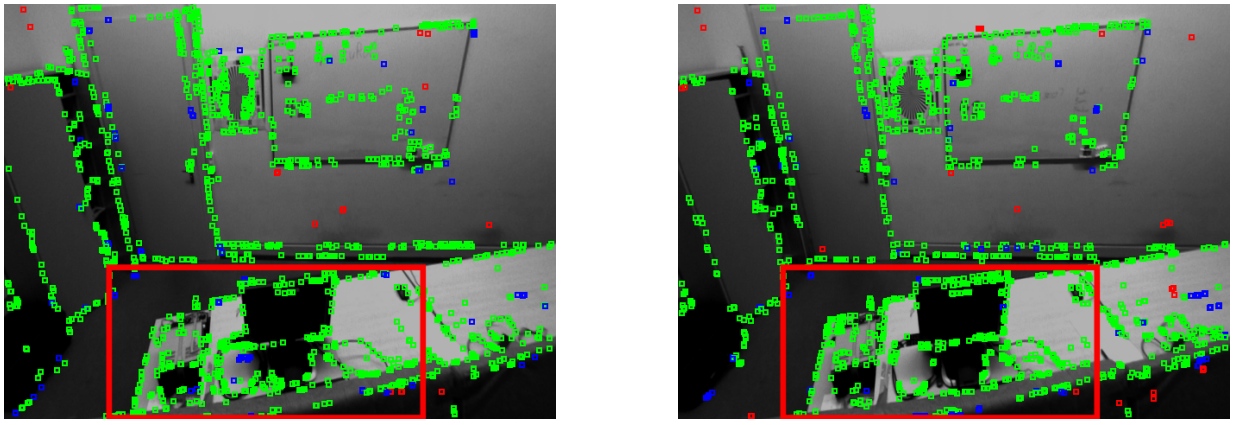


(a) Saliency filtered image using attract scheme

(b) Saliency filtered image using avoid scheme

Figure 3.10: Comparison on saliency weights affected by the attract and avoid deblur methods

To show the difference in point sampling using Deblurred Salient DSO implementation in either attract or avoid scheme Figure 3.11 is shown below. The red rectangle highlights the changes between the two methods. In the attraction scheme, more points are selected in this area.



(a) Saliency filtered image using attract scheme

(b) Saliency filtered image using avoid scheme

Figure 3.11: Comparison of using the two methods. The difference in the sampled points is highlighted in the red rectangle.

The Deblurred Salient DSO implementation enhances DSO by preprocessing the data with an image deblurring network, a saliency predictor network, and an image segmentation network to improve image tracking accuracy. This addition to DSO only adds a minor computational

burden on the system and does not noticeably affect the run-time of the system. The technique presented computes a pixel-wise weight map and attempts two methods of point selection: avoiding and favoring deblurred areas. Examples from the EuRoC MAV dataset illustrate how this affects the final saliency weights and how it influences the point sampling. The next chapter will test and evaluate the implementation similar and compare the performance in accuracy to the DSO, SalientDSO, and DM-VIO implementations.

Chapter 4

Testing

4.1 Test description

4.1.1 Dataset

This implementation will run on the EuRoC MAV dataset. The dataset was created to assess the visual-inertial SLAM and 3D reconstruction capabilities of MAVs. The dataset consists of eleven segments, five of which are in a machine hall, and the remaining six in an office room. [36] The segments are listed in Table 4.1.

Table 4.1: EuRoC dataset, all eleven segments [36]

Dataset	Author comment:
Machine Hall 01	easy
Machine Hall 02	easy
Machine Hall 03	medium
Machine Hall 04	difficult
Machine Hall 05	difficult
Vicon Room 1 01	easy
Vicon Room 1 02	medium
Vicon Room 1 03	difficult
Vicon Room 2 01	easy
Vicon Room 2 02	medium
Vicon Room 2 03	difficult

As shown in the table, there are three different comments to the sequences. The *easy* segments are characterized by slow and steady movement in a decently well-illuminated room. The MAV avoids pointing straight forward into areas that will overexpose the sensor such as a window. The *medium* segments offer faster movement and the illumination has been slightly reduced. Lastly, the *difficult* sequences have to do tracking in low-light conditions, increasing

the effects of motion blur together with a faster movement of the MAV. Additionally in the difficult sequences, the MAV does not face the room as often as in the easy sequences. This results in images with less texture which makes it more difficult to do tracking. An example from all three difficulties in the Vicon Room 2 is shown below in Figure 4.1.

(a) Easy difficulty from the EuRoC MAV dataset

(b) Medium difficulty from the EuRoC MAV dataset

(c) High difficulty from the EuRoC MAV dataset

Figure 4.1: Illustration of the different difficulty levels in the EuRoC MAV dataset. From left to right, the difficulties increase from easy to medium and finally to high.

To provide a comprehensive evaluation, the dataset is run identically to DSO in [15]. All sequences for each camera are run ten times, five of which are played forward and the remaining five are played backward. Since the EuRoC dataset uses a stereo setup and this implementation is monocular, each left and right image are run separately. For example, the sequence "Vicon Room 1 01" is run 20 times, where ten times are for the left camera and the remaining ten times are for the right camera. In total this yields 220 runs for the entire EuRoC dataset.

4.1.2 Absolute trajectory error

For V-SLAM and visual odometry systems, the global consistency of the estimated trajectory is an important quality, and even more so in the former case. By comparing the absolute distances between the estimated and ground truth trajectories, the global consistency can be evaluated. However, since both coordinate frames, the camera, and ground truth trajectories, are defined in arbitrary coordinate frames they first need to be aligned. [37] This is usually referred to as Absolute Trajectory Error (ATE) with SIM(3) alignment. To compute the ATE, the main steps are: Align the estimated trajectory with the ground truth using a similarity transformation, compute the differences between the aligned estimated and ground truth poses, then calculate the RMSE of the differences in the translational components. To perform the alignment, Umeyama's method in [38] is used. Umeyama's method is an extension of Horns method in the original paper "Closed-form solution of absolute orientation using unit quaternions". The extension of Horn incorporates scaling which is particularly useful in the context of monocular visual odometry such as this project. [39]
We have the ground truth trajectory as a set of poses $P_{gt_1}, P_{gt_2}, ..., P_{gt_n}$ and the estimated trajectory as $P_{est_1}, P_{est_2}, ..., P_{est_n}$. First to find the similarity transformation S that best aligns the estimated and ground truth trajectories:

$$S = \min_{S} \sum_{i=1}^{n} ||P_{gt_i} - S \cdot P_{est_i}||^2 \tag{4.1}$$

Then after finding the optimal S, the estimated trajectory is aligned, applying the transformation to each pose:

$$P_{aligned_i} = S \cdot P_{est_i} \tag{4.2}$$

Finally, the ATE is calculated as the RMSE (Rooted Mean Squared Error) of the translational components of the differences:

$$ATE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} ||trans(P_{gt_i}) - trans(P_{aligned_i})||^2} \tag{4.3}$$

It is usually sufficient to only consider the translational component since errors in the rotation will show up as translational errors in subsequent frames[37].

4.2 Results

The results for the EuRoC dataset are presented in Figure 4.2. The graph shows the results for six different runs. Three of the runs are used as a baseline method, namely: "DSO Baseline", SalientDSO Baseline", "DM-VIO", and "SalientDSO Deblurred Data". These types of runs consist of original implementations and what this project's results will be compared against which are the "Deblur Attract" and "Deblur Avoid" methods. The results are shown in Figure 4.2 as a cumulative error plot scaled with the 220 runs. This figure gives a good indication of the overall performance of a method and how it compares to the other. A better implementation will lean towards the left in the graph, and a perfect implementation would be a vertical line at 0.00m on the x-axis.

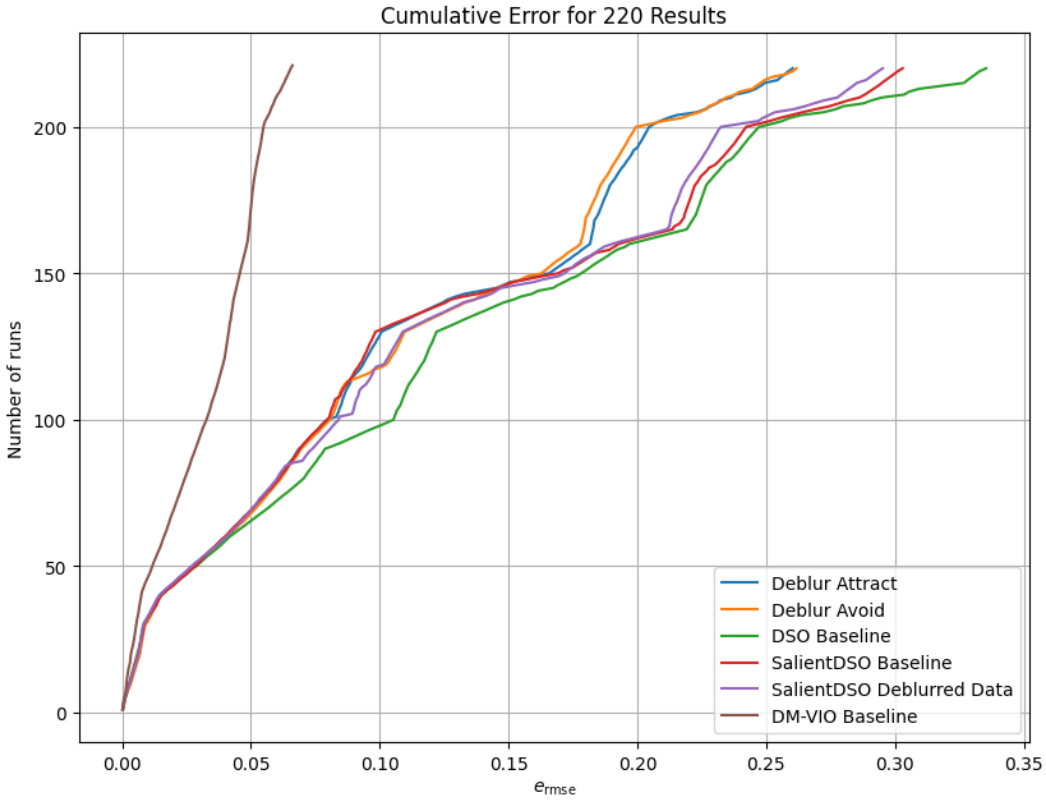


Figure 4.2: Cumulative ATE plot showing the scaled ATE for the 220 runs on the EuRoC dataset.

To better evaluate the result on each type of sequence for each implementation the results have been summarized for each sequence in Table 4.2. The values in the table are the average ATE of the 20 runs for each sequence. The last column *Avg* is the average ATE across all sequences.

Table 4.2: Evaluation of the different methods on the EuRoC MAV dataset. The baseline methods are presented in the first four rows starting with the original implementations of DSO and SalientDSO. SalientDSO Deblur is the original SalientDSO running with deblurred data. DM-VIO is the state-of-the-art visual-inertial odometry method, its accuracy far exceeds any of the other methods. The results for this implementation are shown in the last two rows for the attract and avoid scheme. All the results are in meters

Sequences:	MH1	MH2	MH3	MH4	MH5	V1	V12	V13	V21	V22	V23	**Avg**
DSO	0.065	0.098	0.295	0.317	0.383	0.130	0.339	0.538	0.328	0.226	0.970	**0.335**
SalientDSO	0.069	0.097	0.273	0.232	0.205	0.148	0.357	0.738	0.325	0.220	0.668	**0.303**
SalientDSO Deblur	0.070	0.089	0.287	0.216	0.263	0.202	0.330	0.636	0.305	0.160	0.691	**0.295**
DM-VIO	0.037	0.045	0.097	0.091	0.092	0.074	0.040	0.059	0.027	0.045	0.117	**0.066**
Deblur Attract	0.071	0.089	0.282	0.230	0.202	0.162	0.333	0.629	0.085	0.168	0.613	**0.260**
Deblur Avoid	0.076	0.092	0.281	0.229	0.210	0.251	0.326	0.492	0.085	0.152	0.687	**0.262**

DSO Baseline

The results for DSO are comparable to the original results described in their paper. The answers are not exact as the tools to generate the original plots were not available so they have been manually re-created. Additionally, when processing the EuRoC dataset Engel et. al crops the beginning and end for each trajectory such that the data only consist of images where the MAV is in the air. The exact cropping location was not disclosed in the original paper which may contribute to why the results are not exactly identical but still comparable to the paper.

SalientDSO Baseline

The second baseline method is SalientDSO and shows results very similar to the baseline DSO method. The original paper for SalientDSO did not test on the EuRoC MAV dataset, therefore it is difficult to comment on the performance compared to the original. However, the results do line up with the general performance of SalientDSO which in the original paper was slightly better than DSO. Note that a main strong point for SalientDSO was that it required fewer points to do accurate pose estimation, and they found that with an extremely low point density $N_p = 40$, they still achieved successful tracking.

SalientDSO Deblurred Data

To validate that the extra weighting factor introduced in equations 3.14 and 3.13, the SalientDSO was also run with the deblurred images. The results showed slightly better performance than running on the original dataset without preprocessing.

Deblur Attract and Deblur Avoid

These two runs are a result of the implementation of the project. As seen in the results Figure 4.2, all runs maintain relatively equal accuracy. Around run 70 the different methods begin to diverge. This can largely be explained that the difficult sequences were run between run 60-100 and again at 140-160, and 200-220. In the Tabe 4.2, this corresponds to the columns named MH5, V13, and V23. With the characteristics of

the difficult sequences, it is also expected that these methods would obtain higher accuracy than DSO and SalientDSO.

DM-VIO Baseline

Note on this implementation. The DM-VIO does not support reverse playback while utilizing the IMU data. The runs that DM-VIO should have played in reverse have just been played as a normal sequence. The results shown here are also comparable to the original implementation. The original DM-VIO paper reported an average ATE of 0.069m on the EuRoC MAV dataset [14]. Which is very similar to 0.066m, as observed in Table 4.2.

Chapter 5

Discussion

This chapter will discuss the results presented in Chapter 4 in combination with the implementation presented in Chapter 3. Furthermore, the implementation will be judged in correspondence with the success criteria presented in Table 2.2. Finally, a conclusion that will answer the final problem formulation will be given.

5.1 Results

5.1.1 Baseline methods

The baseline methods were run in order to comment on the performance of this implementation. DSO achieved better results than in the original paper. However, this could be due to the cropping location of the video were not entirely similar. Additionally, the original paper for DSO does not disclose which method is used for the similarity transform. This project used the method presented by Umeyama in 1991, however, the original paper could have used a different method such as Horn's method which may yield a slightly different similarity transform. Although the results are not exact they are deemed good enough for comparison partly because they are performing better than the original paper which may further validate the approach presented in this project. DSO's results follow the expectation of low ATE on the easier sequences while increasing ATE on the more difficult sequences. The only exception is between V21 and V22 in Table 4.2 where the tracking achieved better results on the medium sequence V22 instead of the easy sequence in V21.

The SalientDSO was expected to perform better than DSO which it does, but with a thin margin. To be exact an overall average RMSE of 0.303m compared to DSO's 0.335m is shown. As mentioned in the presentation of SalientDSO in Section 3.3, this project was unable to run the original saliency predictor network (SalGAN) and the image segmentation network (PSPNet). They were replaced with the newer approaches TranSalNet and InternImage 2.5. Since the newer approaches are state-of-the-art models for their respective tasks, this is expected to

either keep the same performance of SalientDSO or improve it. The main difference between DSO and SalientDSO is seen in the Machine Hall 04, and Machine Hall 05 (MH4, MH5) sequences. Those two sequences are also labeled as difficult which highlights the importance of finding good points when the conditions are more ill-conditioned for tracking.

SalientDSO was also run with the deblurred data as input but without the additional weighing of the blurred areas in the point sampling strategy. This was done to confirm that the improvement seen in the proposed Deblurred Salient DSO was not caused by simply using denoised images for tracking. As expected this version performed slightly better than SalientDSO without pre-processed data, with an avg ATE of 0.295m compared to 0.303m as shown in Table 4.2. The improvement in tracking is assumed to originate from the combination of photometric consistency assumption in DSO and that deblurring with the MAXIM model maximizes the signal-to-noise ratio for the image resulting in a more accurate irradiance image.

The top performer was the Visual Inertial system DM-VIO by Usenko et al. [14] It finished with an average ATE of 0.066m followed by the proposed Deblur Salient DSO's ATE of 0.260m, 0.262m. The inclusion of DM-VIO in this comparison was thought out, as the system addresses similar challenges through different methods. The process of image deblurring aids in pose tracking during rapid camera movements, which often result in significant blur [23]. Concurrently, the high-frequency measurements provided by the IMU (Inertial Measurement Unit) ensure precise short-term estimates [35]. These aspects provide a compelling comparison to evaluate the effectiveness of our proposed system. However, it was shown that the proposed system fell behind DM-VIO by a relatively large margin.

Finally, the Deblur Avoid and Deblur Attract presented in this paper showed an increase in accuracy above both SalientDSO with or without using deblurred tracking frames and the DSO baseline methods. The result shown in the cumulative ATE plot in Figure 4.2, shows divergence during the difficult sequences and especially during the Vicon Room sequences. It is interesting that both the Avoid and Attract perform very similarly even though the methods are sort of opposite of each other. The differences in the saliency-filtered image in Figure 3.10 show two very distinct weight maps used for the point selection strategy. In the attract scheme, the reason we get a higher accuracy could be by utilizing the MAXIM model's ability to increase the signal-to-noise ratio and favoring the areas which MAXIM has touched. Another factor may be that the resulting weight map also favors lines and corners. This preference is noteworthy, as utilizing corners and lines is a widely adopted method to do tracking.
Conversely, in the Avoid strategy, the weight of corners and lines is reduced because these areas are purportedly more impacted by motion blur as detected by the MAXIM model. By sampling from areas containing less noise, one would anticipate an increase in accuracy. However, the trade-off is the diminished sampling from lines and corners, which might conversely degrade the tracking result. Regardless, the methods presented cannot follow the accuracy presented by the tightly coupled visual inertial approach in DM-VIO. This indicates that the IMU does not

only improve tracking during motion-blurred frames but improves the overall result for other factors as well. One of those factors could be cases of degenerate motion. The direct formulation does not perform well for movements that consist of pure rotations [15]. Another factor is that the IMU is also included to help with the scale estimate in DM-VIO. The measurements from the IMU can supply the optimization with the size of the movement performed thus approximating a more correct scale, which is important for an accurate trajectory estimation [14]. Using an IMU increases the temporal resolution from camera-only tracking. The movement that happens between keyframes can be more accurately estimated with an IMU.

5.2 Success criterions

The success criteria listed in Table 2.2 guided this implementation. They will be discussed here regarding their success or failure.

Criterion 1 and 2

Criterion number 1 and 2 were related to developing a framework and validating the functionality with simple test cases which have been achieved as documented in the Implementation chapter.

Criterion 3

Criterion number 3 was to evaluate the system's performance against well-recognized public benchmark datasets. The system has been thoroughly evaluated against the EuRoC MAV dataset. This criterion is deemed successful.

Criterion 4

Criterion number 4 was to achieve improved tracking accuracy in comparison to the baseline methods. This criterion has been partly successful. In Table 4.2 the proposed system achieves better performance than DSO and SalientDSO however it falls behind the DM-VIO method. Reasons for this were discussed in section 5.1.1.

Criterion 5

Criterion number 5 was to compare the proposed method with a deblurring method do a visual-inertial odometry method. This was to evaluate the impact of deblurring compared to a IMU as they solve some of the same problems in visual odometry. The results found that only relying on deblurring does increase tracking performance however, there is much more to be gained by using an IMU. This criterion is deemed successful as an answer has been found to how IMU and deblurring compare.

Criterion 6

Criterion number 6 was to assess the computational efficiency of the hybrid system. The efficiency should operate within an acceptable computational time and resources usage. The computational time of this approach has not been thoroughly evaluated, however, from preliminary testing it is found that the avoid or attract blur does not increase the runtime of DSO noticeably, however, the preprocessing does not allow for real-time operation. As stated in the description of the preprocessing in section 3.1, the combined processing time for each image exceeds several seconds.

Criterion 7

Criterion number 7 was to analyze failure cases of the developed framework to learn about the limitations of the system. This criterion has not been explored and is therefore not deemed successful. It would be a sensible next step for future work to better understand the systems performed.

5.3 Future work

Looking ahead, there are several promising directions one could take to further advance this method. Therefore four points have been listed, any of the four points could be the next step.

1. **Testing on multiple datasets:** While the 3. criterion was deemed successful, testing on various datasets could provide valuable insights. Doing so would explore the performance of the developed system in other environments. This could also help with Criteria 7 which was to analyze failure cases, by exposing the system to scenes it might fail to track. Testing on the photometrically calibrated monocular visual odometry dataset developed by Engel et al. [33] would be a sensible next step for the project. This would also make greater use of the photometric calibration formulation used explained in section 3.2.1. Testing on a photometrically calibrated dataset is essential to compare the performance with indirect or deep learning methods which should also be done subsequently.

2. **Parameter study**. This project employed a avoid or a attract scheme to deblurred portions of the image. A more elaborate parameter study could potentially figure out which of these methods would perform better and with more convincing reasons as to why.

3. **Investigate other pre-processing methods** Since the project did not achieve performance comparable to introducing an IMU unit into the system, other pre-processing methods could be interesting to investigate. The MAXIM model can also do other image processing tasks such as denoising, dehazing, exposure correction and image enhancement. A combined effect of all these factors could potentially further increase the performance of the system.

4. **Unsuccessful criteria**. This project met most of the success criteria, but some were not fully realized. Mainly criterion 7 which concerned the analysis of failure cases was not explored. Further work could investigate this point which may provide valuable insights for future improvements and combine the efforts of other pre-processing methods, and a parameter study. Additionally, for practical applications further work could also focus on improving computational efficiency to meet criterion 6.

To summarize, the implementation and results of this project have proven promising, and there are as described, opportunities for future work to further enhance the system's performance and its applicability.

5.4 Conclusion

This project was guided by the following final problem statement:

"How can the integration of deep learning enhancements augment direct visual odometry to better handle rapid camera movements and adverse imaging conditions?"

After conducting an exhaustive analysis of the challenges in visual odometry and alternative methods to visual odometry, this project landed on combining saliency-filtered images with a deblurring method to obtain stronger point samples serving as input to DSO. The proposed system was benchmarked against baseline methods for direct visual odometry namely DSO and SalientDSO. Additionally, the system was evaluated against the state-of-the-art visual-inertial method DM-VIO since both the deblurring module and the IMU attempt to resolve similar issues regarding camera motion. All methods were comprehensively evaluated on the EuRoC MAV dataset. The proposed system achieved better compared to the baseline direct visual odometry methods results especially on difficult sequences. An average ATE of 0.26m across the dataset for the proposed system compared to 0.335m for DSO and 0.303m for SalientDSO. In short, the final problem formulation has been answered. Integrating a deblurring module together with salient point sampling has proven to increase trajectory estimation accuracy during blurry frames caused by rapid camera movements or long exposure times in low-illuminated conditions. Further research proposes several directions. One direction is to investigate other pre-processing methods such as image denoising, dehazing, and image enhancement to improve tracking. Another aspect is to test the system for failure cases, this could for example be done using other benchmark datasets. In conclusion, the implementation and the results of this project have proven promising and there are as described opportunities to further enhance the system.

Bibliography

[1] Navvis. *Reality Capture*. Accessed: 2023-05-31. 2023. URL: https://www.navvis.com/reality-capture.

[2] David Scaradozzi, Silvia Zingaretti, and Arianna Ferrari. "Simultaneous localization and mapping (SLAM) robotics techniques: a possible application in surgery". In: *Shanghai Chest* 2 (1 Jan. 2018). ISSN: 2521-3768. DOI: 10.21037/SHC.2018.01.01. URL: https://shc.amegroups.com/article/view/4083/htmlhttps://shc.amegroups.com/article/view/4083.

[3] Abhishek Singh. *Super Mario Bros Recreated as Life Size Augmented Reality Game*. [Online; accessed on 1-June-2023]. 2017. URL: https://www.youtube.com/watch?v=QN95nNDtxjo.

[4] Paweł Nowacki and Marek Woda. "Capabilities of ARCore and ARKit Platforms for AR/VR Applications". In: *Engineering in Dependability of Computer Systems and Networks*. Ed. by Wojciech Zamojski et al. Cham: Springer International Publishing, 2020, pp. 358–370. ISBN: 978-3-030-19501-4.

[5] Sungchul Hong et al. "Visual SLAM-Based Robotic Mapping Method for Planetary Construction". In: *Sensors* 21.22 (2021), p. 7715. DOI: 10.3390/s21227715. URL: https://doi.org/10.3390/s21227715.

[6] Davison. "Real-time simultaneous localisation and mapping with a single camera". In: *Proceedings Ninth IEEE International Conference on Computer Vision*. 2003, 1403–1410 vol.2. DOI: 10.1109/ICCV.2003.1238654.

[7] Iman Abaspur Kazerouni et al. "A survey of state-of-the-art on visual SLAM". In: *Expert Systems With Applications* 143 (2021), p. 103778.

[8] A. Dias A. Martins J. Almeida A. Moura J. Antunes. "Graph-SLAM Approach for Indoor UAV Localization in Warehouse Logistics Applications". In: *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)* (Apr. 2021).

[9] Jongseok Lee et al. "Virtual Reality via Object Pose Estimation and Active Learning: Realizing Telepresence Robots with Aerial Manipulation Capabilities". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)* (). URL: https://www.youtube.com/watch?v=JRnPIARW8xY.

[10] Fazil E. Uslu et al. “Engineered Extracellular Matrices with Integrated Wireless Microactuators to Study Mechanobiology”. In: *Advanced Materials* 33 (40 Oct. 2021). ISSN: 15214095. DOI: 10.1002/ADMA.202102641.

[11] Yong Dai, Jiaxin Wu, Duo Wang, et al. “A Review of Common Techniques for Visual Simultaneous Localization and Mapping”. In: *Journal of Robotics* 2023 (2023).

[12] Raúl Mur-Artal and Juan D. Tardós. “ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras”. In: *IEEE Transactions on Robotics* 33.5 (2017), pp. 1255–1262. DOI: 10.1109/TRO.2017.2705103.

[13] Giorgio Grisetti et al. “A tutorial on graph-based SLAM”. In: *IEEE Intelligent Transportation Systems Magazine* 2.4 (2010), pp. 31–43. DOI: 10.1109/MITS.2010.939925.

[14] Lukas von Stumberg and Daniel Cremers. “DM-VIO: Delayed Marginalization Visual-Inertial Odometry”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 1408–1415. DOI: 10.1109/LRA.2021.3140129.

[15] Jakob Engel, Vladlen Koltun, and Daniel Cremers. “Direct sparse odometry”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.3 (2016), pp. 611–625.

[16] 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)*. IEEE. 2018, pp. 2510–2517.

[17] Zhelin Yu, Lidong Zhu, and Guoyu Lu. “VINS-Motion”. In: *IEEE International Conference on Robotics and Automation* (2021). URL: https://ieeexplore-ieee-org.zorac.aub.aau.dk/stamp/stamp.jsp?tp=&arnumber=9562103.

[18] Vladyslav Usenko et al. “Visual-Inertial Mapping With Non-Linear Factor Recovery”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 422–429. DOI: 10.1109/LRA.2019.2961227.

[19] Sen Wang et al. “Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks”. In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2017, pp. 2043–2050.

[20] Rujun Song et al. “ContextAVO: Local context guided and refining poses for deep visual odometry”. In: *Neurocomputing* 533 (2023), pp. 86–103.

[21] Huai-Jen Liang et al. “SalientDSO: Bringing Attention to Direct Sparse Odometry”. In: *IEEE Transactions on Automation Science and Engineering* 16.4 (2019), pp. 1619–1626. DOI: 10.1109/TASE.2019.2900980.

[22] Jiandong Guo, Rongrong Ni, and Yao Zhao. “DeblurSLAM: A Novel Visual SLAM System Robust in Blurring Scene”. In: *2021 IEEE 7th International Conference on Virtual Reality (ICVR)*. 2021, pp. 62–68. DOI: 10.1109/ICVR51878.2021.9483818.

[23] Peidong Liu et al. “MBA-VO: Motion blur aware visual odometry”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 5550–5559.

[24] Davide Scaramuzza and Friedrich Fraundorfer. "Visual odometry [tutorial]". In: *IEEE robotics & automation magazine* 18.4 (2011), pp. 80–92.

[25] M. O. A. Aqel, M. H. Marhaban, M. I. Saripan, et al. "Review of visual odometry: types, approaches, challenges, and applications". In: *SpringerPlus* 5.1 (2016), p. 1897. DOI: 10.1186/s40064-016-3573-7.

[26] Ke Wang et al. "Approaches, challenges, and applications for deep visual odometry: Toward complicated and emerging areas". In: *IEEE Transactions on Cognitive and Developmental Systems* 14.1 (2020), pp. 35–49.

[27] Johan Vertens, Abhinav Valada, and Wolfram Burgard. "SMSnet: Semantic motion segmentation using deep convolutional neural networks". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 582–589. DOI: 10.1109/IROS.2017.8202211.

[28] Qingxi Zeng et al. "Robust Mono Visual-Inertial Odometry Using Sparse Optical Flow With Edge Detection". In: *IEEE Sensors Journal* 22.6 (2021), pp. 5260–5269.

[29] Jianxun Lou et al. "TranSalNet: Towards perceptually relevant visual saliency prediction". In: *Neurocomputing* 494 (2022), pp. 455–467.

[30] Zhengzhong Tu et al. "Maxim: Multi-axis mlp for image processing". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 5769–5780.

[31] Wenhai Wang et al. "Internimage: Exploring large-scale vision foundation models with deformable convolutions". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 14408–14419.

[32] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. 2nd ed. Cambridge University Press, 2004. DOI: 10.1017/CBO9780511811685.

[33] Jakob Engel, Vladyslav Usenko, and Daniel Cremers. "A photometrically calibrated benchmark for monocular visual odometry". In: *arXiv preprint arXiv:1607.02555* (2016).

[34] Michael D Grossberg and Shree K Nayar. "What is the space of camera response functions?" In: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* Vol. 2. IEEE. 2003, pp. II–602.

[35] Stefan Leutenegger et al. "Keyframe-Based Visual-Inertial Odometry Using Nonlinear Optimization". In: *The International Journal of Robotics Research* 34 (Feb. 2014). DOI: 10.1177/0278364914554813.

[36] Michael Burri et al. "The EuRoC micro aerial vehicle datasets". In: *The International Journal of Robotics Research* (2016). DOI: 10.1177/0278364915620033. eprint: http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.full.pdf+html. URL: http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.abstract.

[37] Jürgen Sturm et al. "A benchmark for the evaluation of RGB-D SLAM systems". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012, pp. 573–580. DOI: 10.1109/IROS.2012.6385773.

[38] S. Umeyama. “Least-squares estimation of transformation parameters between two point patterns”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.4 (1991), pp. 376–380. DOI: 10.1109/34.88573.

[39] Zichao Zhang and Davide Scaramuzza. “A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 7244–7251.