Visual-Inertial Odometry, Mapping and Re-Localization through Learning

Ronald Clark

Exeter College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2017

"The cosmos is full beyond measure of elegant truths; of exquisite interrelationships; of the awesome machinery of nature" - Carl Sagan

Statement of Authorship

This thesis is submitted to the Department of Computer Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, except where otherwise stated.

Ronald Clark, Exeter College

Funding

The work described in this thesis was funded by the Engineering and Physical Sciences Research Council (EPSRC).

Acknowledgements

I would like to thank Wilma Clark and Neil Clark for the helpful discussions as well as Sen Wang for assisting with the experiments and Shuyu Lin for proofreading.

Abstract

Precise pose information is a fundamental prerequisite for numerous applications in robotics, AI and mobile computing. Monocular cameras are the ideal sensor for this purpose - they are cheap, lightweight and ubiquitous. As such, monocular visual localization is widely regarded as a cornerstone requirement of machine perception. However, a large gap still exists between the performance that these applications require and that which is achievable through existing monocular perception algorithms.

In this thesis we directly tackle the issue of robust egocentric visual localization and mapping through a data-centric approach. As a first major contribution we propose novel learnt models for visual odometry which form the basis of the ego-motion estimates used in later chapters. The proposed approaches are less fragile and much more robust than existing approaches. We present experimental evidence that these approaches can not only approach the accuracy of standard methods but in many cases also show major improvements in computational and memory efficiency.

To cope with the drift inherent to the odometry methods, we then introduce a novel learnt spatio-temporal model for performing global relocalization updates. The proposed approach allows one to efficiently infer the global location of an image stream at the fraction of the time of traditional feature-based approaches with minimal loss in localization accuracy.

Finally, we present a novel SLAM system integrating our learnt priors for creating 3D maps from monocular image sequences. The approach is de-

signed to harness multiple input sources, including prior depth and ego-motion estimates and incorporates both loop-closure and relocalization updates. The approach, based on the well-established standard visual-inertial structure-from-motion process, allows us to perform accurate posterior inference of camera poses and scene structure to significantly boost the reconstruction robustness and fidelity.

Through our qualitative and quantitative experimentation on a wide range of datasets, we conclude that the proposed methods can bring accurate visual localization to a wide class of consumer devices and robotic platforms.

Contents

List of Figures

Chapter 1

Introduction

Accurate pose information is a key enabler of mobile robot autonomy and many other applications running on mobile computing platforms. In outdoor environments, global position information is readily available, where GPS has secured itself as the de facto standard. GPS, however, has the major disadvantage that it cannot provide orientation information and its signals are highly attenuated in and around buildings. This has led to a major academic and commercial interest in developing technologies which are able to obtain pose information through other means.

However, it is also possible to perform localization and mapping using sensors found on-board, have the advantage of not relying on external hardware and thus much research has focussed on using self-contained sensors such as cameras and inertial measurement units (IMUs) to provide pose information [79]. Great strides have been made in the accuracy of localization and mapping technologies, however, the current systems still leave much to be desired. The accuracy of infrastructure-free solutions is often inadequate in applications where, for example, sub-meter accuracy is required.

One way to achieve this level of accuracy and flexibility is through camera-based mapping and localization methods that rely on a monocular camera to localize the

device. Cameras are, after-all, becoming cheap and ubiquitous. Camera-based pose estimation and mapping techniques have been extensively explored over the past 30 years (see [55] for a review) where they have enjoyed great success in situations when expert tuning and controlled environments are available. Vision-based localization and mapping, however, face severe challenges when operating in human-centric environments and thus their widespread application has been rather limited up to now. Examples of such challenges include stringent power and processing constraints of real-world platforms, the challenges related to coping with dynamic elements, and of-course the expert tuning required for the algorithms to operate properly. Current visual SLAM methods are fragile and cumbersome [17], requiring map representations tailored to a specific algorithm. This significantly limits the flexibility of existing approaches, and makes it difficult to adapt them to new environments.

A promising means of addressing these issues is through the use of machine learning techniques, particularly *deep learning* [89] which has revolutionized the field of computer vision and is starting to have a significant impact on robotics and machine perception in general. These methods have the ability to learn highly complex models directly from data, thereby capturing the intricacies and complexities of real-world processes in a manner that is impossible to achieve using hand-designed algorithms. For example, deep convolutional neural networks are able to estimate the depth of a scene from a single image [41] and provide semantic labels for images captured under unconstrained conditions - both tasks which would traditionally have required extensive hand-engineering to achieve usable models. However, although deep learning has proven successful for tasks previously out of reach of traditional computer vision techniques, its application in more mathematically-grounded and well-understood areas, such as visual SLAM, is more challenging. Instead of completely replacing well-understood traditional methods. In this regard, the specific challenges that must be overcome include finding a suitable means of fusing the deep network output with

standard geometry-based knowledge and, in-turn, finding a good means of characterizing network uncertainty [17].

In this thesis we take the view of *deep networks as priors* and propose to use deep models as a means of extracting and capturing prior knowledge for the purpose of localizing and tracking mobile platforms. The introduction of the learned priors allows us to not only increase the accuracy of the SLAM process, but also to guide it in areas where traditionally there would be too little information to obtain a reliable output.

Chapter 2

Background

Many different sensor modalities have been investigated to enable mobile robots to localize and navigate where no GPS signals are available. Amongst these, radio-based methods have been quite popular, however, these rely on the installation or presence of external transmitters such as WiFi access points [48], Bluetooth low-energy (BLE) tags [47], Ultra-wideband (UWB) radios [131], or specially installed visible light sources [88]. For mobile robots, high-precision sensors such as LiDAR setups and stereo-rigs have been popular. However, many mobile platforms do not have access to highly specialized cameras such as stereo setups, or on-device laser range finders. As such, camera-based localization techniques that rely on only monocular images are highly desirable and their application has been shown in underwater vehicles, industrial machines, self-parking cars and even planetary rovers.

In this section, we survey and identify state-of-the-art vision-based pose tracking, mapping and localization methods. These techniques have to derive both the user's position and the map using a sequential series of images. Many visual algorithms exist to estimate the pose, or ego-motion, of a device relative to a known starting point. These can be broadly classified into visual odometry (VO) and visual SLAM (vSLAM) algorithms.

2.1 Visual Odometry

VO algorithms specialise in incremental ego-motion of the camera. A general VO algorithm operates by extracting features in an image, matching the features between the current and successive images and then computing the optical flow. The motion can then be computed using the optical flow. This can be done with or without recovering the structure of the scene. For example, in the case of forward motion (such as a person walking), the optical flow diverges from a single point in view of the camera which provides the direction and rotation of the inter-frame motion. Structured methods triangulate feature points between successive frames and perform least-squares optimization to refine the motion in a maximum-a-posteriori sense by minimizing the discrepancies in their re-projections.

The fast semi-direct monocular visual odometry (SVO) algorithm of Forster [52] is an example of a state-of-the-art VO algorithm. It is designed to be fast and robust by operating directly on the image patches, not relying on slow feature extraction. Instead, it uses probabilistic depth filters on patches of the image itself. The depth-filters are initialized at feature points, however, this is only carried out at key-frames. The depth filters are then updated through whole image alignment. This visual odometry algorithm has all the desired characteristics for ego-centric application - it runs on a laptop at around 300 frames per second and at 55 fps on an embedded platform. Its probabilistic formulation, however, makes it difficult to tune and it also requires a bootstrapping procedure to start the process. As expected, its agile performance depends heavily on the hardware used - typically global shutter cameras operating at higher than 50 fps needs to be used to ensure the odometry estimates remain accurate.

2.2 Visual SLAM

Even the state-of-the-art visual odometry algorithms exhibit considerable drift over time. Visual simultaneous localization and mapping (vSLAM) algorithms thus build an on-line visual map of the world while simultaneously estimating the camera ego-motion to further reduce drift. vSLAM systems include a "place" recognition component to detect loop closures and re-localize when the system detects that it is lost. Loop-closure methods recognise previously visited places, "closing the loop" of poses and distributing accumulated errors across the entire path.

A number of competing state-of-the-art vSLAM systems exist. ORB-SLAM [113] is a feature-based SLAM method based on ORB features. By establishing map point correspondences using the ORB descriptor, ORB-SLAM is robust to occlusions and fast motions without requiring additional sensory input such as an IMU. It locally optimizes the keyframe poses and map points through a bundle adjustment procedure and detects loop closures using DBoW2 for appearance-based place recognition. SVDO [45] is a (semi) dense method for visual pose estimation from a monocular camera. It uses a method similar to that employed by RGB-D slam systems. It first estimates a depth map using a sequence of frames and then aligns subsequent depth maps to form a coherent model. The map or model produced by the SVDO system is very rich and easily interpretable as it retains most of the texture from the original video. It is more computationally intensive compared to odometry methods, but can operate at $> 10Hz$ on a desktop PC. The authors of the SVDO algorithm have demonstrated its ability to run on a resource-constrained cellphone system, although this was with a limited map size.

vSLAM algorithms, however, are by no means perfect as the created map often becomes inconsistent, i.e the system does not know it is lost - causing it to lose track of the current pose. This leads to the significant disadvantage that whenever SLAM algorithm gets lost, map corruption gradually occurs and the stored feature map

have to be re-initialized. In situations where it is likely that the system will often get lost, relative pose estimates from a VO system are more desirable. Furthermore, most current visual localization techniques build homogenous maps consisting of specialised features. For example, ORB-SLAM builds maps of sparse ORB feature points [111], LSD-SLAM [45] on the other hand build maps consisting of the edge regions in the image with highest gradient magnitude.

2.3 Constrained Visual SLAM and VO

Regardless of the algorithm, both single camera vSLAM and VO solutions are unable to observe the scale of the scene and will always be subject to both scale drift and a scale ambiguity. To recover correct scale drift and recover metric scale, additional information needs to be introduced. This can be accomplished through either loop-closure, as used in vSLAM systems, or through a *constrained* SLAM solution. Although they mitigate drift, loop-closures do not give correct metric scale - this can only be achieved using constrained SLAM solutions.

Constrained SLAM methods introduce additional information to constrain the solution and recover metric scale. The additional information can take the form of structure or motion constraints. Structural constraints involve detecting and identifying structures in the scene which have a known size or dimension - for example, many traditional solutions use the height of the camera from the ground-plane to recover and correct scale. More recent methods combine object recognition with visual SLAM [21, 124, 132] to produce a map of the objects along with a scale-corrected trajectory. The authors show that not only can the scale ambiguity be resolved by the object detection, but the object detection performance can also be improved by making use of the pose estimates obtained from the SLAM.

Constrained SLAM can also use motion constraints which is sourced from sup-

porting hardware - usually in the form of an IMU. In terms of pose estimation, the information available from an IMU and that which can be obtained from visual methods exhibit a unique relationship. Positional information is unobservable using either of these modalities in isolation. In theory, the complementary nature between inertial measurements from an IMU and visual data should enable highly accurate ego-motion estimation under any circumstances: visual localization techniques are entirely reliant on observing distinctive features in the environment and in indoor situations these techniques are plagued by intermittent occlusions. On the other hand, the pose of an IMU device can be easily tracked through double integration of the acceleration data, providing relative position estimates when the visual information is unavailable. In state-of-the-art systems, this tightly-coupled data fusion is achieved through a Multi-state Constraint Kalman Filter (MSCKF) [109] or a Sliding Window Filter (SWF) [25] and a combination of visual and inertial data in this manner for the purpose of pose estimation is known as tightly-coupled visual-inertial-odometry (VIO).

An example of a state-of-the-art consumer device which is able to fully utilize the complementary nature of the inertial and visual sensors is Google's *Project Tango*. Project Tango devices include a gyroscope, accelerometer and camera to estimate 6-DoF motion tracking, enabling users to track 3D motion of the device while simultaneously creating a map of the environment. Its pose tracking is performed by VIO. Tango's odometry is computed by fusing the visual and inertial data using an MSCKF, SWF or similar. Project Tango can be considered an *engineered* VIO system in that it uses specific hardware components tailored to the task of reliably performing pose tracking in unconstrained environments. The particular hardware components that it employs to achieve this include 1) a wide angle (170°) global shutter camera, 2) synchronized IMU and camera timestamps and 3) a depth sensor. As with other state-of-the-art VIO systems [120], it achieves a drift rate of around 1% of the distance travelled. To perform localization, the device includes an "area-learning" capability

which stores BRISK, FREAK or SURF descriptors captured along the trajectory. A place recognition algorithm is then used to perform re-localization [99].

2.4 Mapping and scene representations

There are two main types of maps used for visual localization - metric maps and topological maps.

Topological maps represent the environment in a relative manner by capturing sequences of places. The places themselves are only defined in a very coarse manner. Topological maps are typically represented using graph-based structures where the nodes correspond to places and edges describe the approximate transformations between places.

Rivera et al. [128] define a topological map consisting of "visual paths" to associate locations captured by wearable cameras for the purpose of indoor navigation. They further show that considering multiple frames (as opposed to just single frames in isolation) improves the localization performance across the visual paths.

Topological maps often used in teach-and-repeat navigation methods have been effective in navigating robots where it is difficult to obtain globally accurate metric maps such as in mines as well as indoors. These visual navigation methods have the advantage of minimal set-up time. In the first traverse of the environment, the mapper records the visual information that it observes along the trajectory. In the navigation phase, the follower is able to localize itself along the trajectory of the mapper and follow the path to the desired location. The system in [56] creates a relative map of stereo images allowing a follower to localize at any location along the map, navigation to any destination can be performed.

Metric maps aim to establish an accurate correspondence between an environment and its representation to allow for precise localization and path planning. The maps

that take the form of 3D models or 3D point clouds are examples of metric maps. Such point-cloud maps can be obtained from an unordered set of images using Structure from motion algorithms. Structure from motion algorithms are characterized by the offline reconstruction of 3D maps from a series of unordered images. These algorithms operate on image pairs in a greedy manner, gradually building up a consistent model of the 3D scenes. Through bundle adjustment (i.e. a non-linear optimization on the re-projection errors of the recovered points), SfM techniques produce a model output of highest possible accuracy. From a user perspective, the camera intrinsic parameters do not need to be determined prior to performing the reconstruction, as these parameters can be included in the optimization process. For this reason SfM algorithms have been popular in the indoor localization literature as methods for constructing 3D models against which the mobile device can be localized.

2.5 Pose estimation and Relocalization

When a 3D model of discriminative feature points is available, such as is obtained using SfM, then the pose of query images can be found using camera-resectioning. This process first establishes feature matches between the image and the 3D model by either 3D-to-2D [167] or 2D-to-3D [135] matching. The camera pose is then found by using RANSAC in combination with a PnP algorithm [137]. Camera localization techniques where direct matching to the 3D model is performed are generally much more computationally expensive compared to appearance-based matching methods and are often infeasible if no prior pruning on the model is conducted.

Recent work in the context of autonomous vehicles [149], investigated the use of a 3D LiDAR map for monocular camera-based localization. As LiDAR and visual images capture vastly different data, their method is based on minimizing the mutual information distance between the LiDAR points projected into the view of a multi-

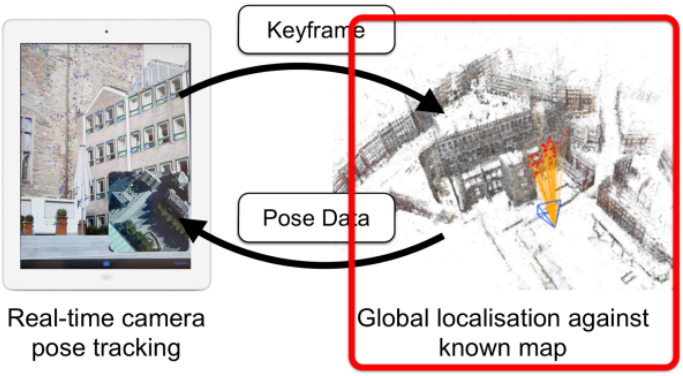


Figure 2.1: Model-based localization using a model made up of sparse features such as those produced by SfM techniques [137].

camera set-up mounted on their vehicle. They further extended this work by using the prior LiDAR point cloud as a means of overcoming the issues associated with performing pose estimation in heavily populated urban environments [102]. Here, they use the point cloud projected into the views of a monocular VO system as a means of performing distraction suppression in the images. By utilising the prior knowledge of the areas appearance, they are able to focus the attention of the pose estimation system only on the stable and static parts of the scene. In this manner, they are able to obtain reliable pose estimates even when the cameras are nearly completely obscured by moving objects. The authors also demonstrate the use of mutual information for aligning a live monocular camera stream to orthographic images created from a stereo camera stream [114].

Appearance-based place recognition, is used for loop-closure in vSLAM systems, and is an important aspect of topological mapping systems, where it is used for localization. Appearance-based localization methods like FAB-MAP [31] and others [119, 146] recognise places purely based on the appearance of the image. In most cases, the image appearance is described by adopting the BoVW model. This process is similar to the large-scale image-retrieval problem. In classical image retrieval, however, the aim is to find as many relevant database images as possible, while appearance-

based localization methods aim at finding co-located images in the database [137]. Thus, approaches such as FAB-MAP are characterised by incredibly high precision with typically low-recall rates. By integrating temporal links between places, as is done in SeqSLAM [106], the recall rate of appearance-based localization methods can be significantly improved. Appearance methods can also be used for localizing in model-based maps by rendering synthetic views at a large number of candidate locations and orientations. This approach of rendering synthetic views was used by [74] as a pre-processing step to speed up the direct-matching process in localization using an SfM model. A recent method [122] uses continuous synthetic view rendering of a dense 3D model to perform localization by minimizing the difference between the query image and rendered view in terms of the normalized information metric.

RatSLAM [107] bridges the gap between metric pose estimation and appearance-based place recognition. RatSLAM uses a hippocampus-inspired model which integrates visual observations of landmarks and odometric measurements using a neural network consisting of place cells. Sensory input, corresponding to for example left right turns or forward motion, activates specific place cells and the output of the cells is integrated using an attractor network.

2.6 Consumer adoption

Visual SLAM algorithms are slowly being adopted in consumer products, such as illustrated in Figure 2.2. Even so, most areas in everyday-life, such as those prototypical of the places in which ubiquitous localization are likely to be deployed, are overwhelmed by walking people and moving vehicles.

These dynamic elements in the environment pose a significant challenge to both visual odometry and appearance-based localization methods and various methods have been proposed to address the problem of localizing in environments which are

(a) (b) (c)

Figure 2.2: After 30 years of research, visual SLAM solutions are slowly making their way into consumer applications, but many challenges persist. (a) Google Project Tango is the first consumer hardware platform designed specifically for visual SLAM (b) The Dyson 360 Eye was the first consumer device to utilize visual SLAM technology (c) The DJI Mavic Pro relies on visual odometry and GPS for navigation

subject to dynamic change. While fast dynamic elements affect both VO and visual SLAM, illumination and weather changes are mainly only a concern for relocalization methods [105]. The variation of the feature matching performance across different times of day. Current research on coping with these conditions aim at learning to model the change in appearance. [126] learns the matching function for local descriptors with the goal of creating a matching function which captures the variance due to illumination change. A more recent method [20] learns the actual local feature descriptors, by projecting image patches onto a low-dimensional space which is discriminative between matching and non-matching pairs, but invariant to the lighting conditions. Milford et al. [105] present a more complex approach. They use a two-step algorithm which first matches the image using a global descriptor and then perform a verification step on higher-resolution patches. They furthermore use visual saliency as a means of improving the performance both with respect to the perceptual change and computational speed of their algorithm.

2.7 Challenges

A vast number of standard visual SLAM algorithms have been proposed, each addressing the failure cases of the previous, but introducing new failure cases of their own. These failure cases are often not very intuitive to understand and difficult to eliminate. In this thesis we focus on learning-based approaches which allow us to overcome these failure cases by learning from large amounts of labelled (and unlabelled) data. We focus on a number of specific challenges. The specific high-level challenges which we focus on include:

Accuracy The development of traditional visual localization and mapping systems has centered around achieving increases in accuracy. In this thesis we focus wholly on metric accuracy of the poses and maps and are interested in obtaining full 6-DoF pose estimates which are essential for most AR and VR applications as well as in mobile robotics. The accuracy will be quantified as the difference between the actual position and orientation and those estimated by the system. Ideally a visual localization and mapping system should be metrically accurate to the centimeter level.

Robustness and device diversity The robustness of the system describes its ability to operate in a range of environments, under different conditions. This is another important but neglected area in both visual and general SLAM technologies. Robustness dictates the percentage of time that a localization system is able to reliably localize the robot. This has major implications for applications such as navigation where it is critical that the system be able to localize under any conditions. Another challenge for mobile localization and mapping is to account for the inherent device diversity. This diversity arises both in terms of the camera and other sensors which vary greatly in terms of quality, features and performance. For example, cameras have a range of field-of-views, dynamic

ranges, shutter speeds and sensitivities. Accounting for these are essential to a practical system. The fragility of existing visual SLAM solutions is nicely demonstrated in Table 2.1 (reproduced from [93]) which shows the performance of the standard methods across a variety of datasets, including indoor (In), outdoor (Out) and underwater (UW).

Efficiency and scalability A major challenge for visual mapping and localization on mobile platforms if efficiency. To be efficient, the system must use as few resources as possible, allowing the device to operate as long as possible on limited battery capacity and to perform critical application-related auxiliary tasks such as path planning or object recognition. Related to efficiency is scalibility. One of the major challenges of localization and mapping systems in general, and one which we will specially address in this thesis is how to make it scalable. Scalability here refers to the system's ability to operate across areas of different size, such as across museums and tiny office rooms.

Table 2.1: High-level evaluation of various visual state-estimation methods across indoor (In), outdoor (Out) and underwater (UW) datasets. Qualitative descriptors: red - failure across entire dataset, green - works across entire dataset.

	O1	I1	O2	I2	O3	I3	U1	U2
MonoSLAM [33]								
LSD-SLAM [44]								
SVO [53]								
LibVISO [64]								
PTAM [84]								
RATSLAM [107]								
COLMAP [138]								
ORB-SLAM [112]								

2.8 Research Questions and contributions

In this thesis we aim to create a visual localization and mapping system that does not suffer from the fragility of traditional approaches and that is more robust and suitable for widespread adoption. To achieve this we concentrate on approaches more closely inspired by biology (specifically artificial neural networks). Our hypothesis is three-fold. Firstly, we propose that integrating learning methods can allow our system to harness data to reduce errors through training. Secondly, we hypothesise that considering the camera not as a single sensor in isolation, but also in tandem with complimentary sensors (eg. IMU) we can achieve higher efficiency and accuracy for localization. Finally, the last hypothesis of this thesis is that monocular cameras provide sufficient information for localization and mapping that can be competitive with even well-calibrated stereo and LiDAR setups.

The main research questions that we consider are:

2.8.1 Question 1: How can we harness learning-based approaches to create priors for robust vision-based localization and mapping?

This component of the work focuses on how to derive reliable motion and global relocalization priors from image-sequences and complementary IMU data. In particular we tackle the questions, can we use a data-driven approach to learn to estimate ego-motion from raw sensor streams? Can this be achieved without requiring any expert user tuning? This question addresses the challenge of **robustness and device diversity** for visual localization and mapping.

2.8.2 Question 2: Can we integrate these learning-based priors with standard Visual-SLAM optimization for accurate localization and mapping?

The first question addressed the problem of robustness, and as will be demonstrated, the learning-based models alone do not necessarily achieve state-of-the-art performance compared to standard optimization approaches in terms of accuracy. With this question seeks to utilize the priors in conjuction with a standard SLAM method in order to successfully overcome the challenges associated with performing robust vision-based mapping and localisation. In particular, this component entails integrating the motion priors with an inference method to find a maximum a-posteriori solution for the localization and mapping problem.

2.8.3 Question 3: How can we make visual localization methods more efficient?

Most mobile platforms are subjected to memory and computational constraints. As our visual maps increase in size these constraints become ever more important. Our third research question is related the challenge of localization **efficiency**. Here we seek to perform efficient localization in the maps created using our SLAM system. Emphasis here is to be placed on: 1) finding ways to integrate multi-modal auxiliary sensor data in the reconstruction process, 2) efficiently localizing in the reconstructed map, 3) allowing for the integration of location and motion priors to boost the localization robustness.

2.9 Thesis outline

An outline of the remaining chapters of this thesis is presented in Fig. 2.3 which summarizes how the individual components fit together. Each component is discussed in detail in the relevant chapters.

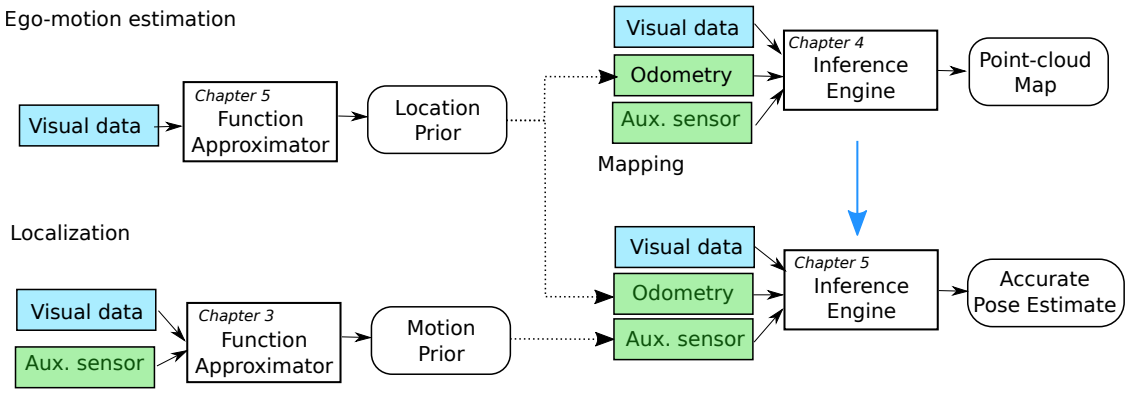


Figure 2.3: Outline of the proposed framework and thesis structure.

Chapter 4 and 5: In Chapter 4 we address the problem posed in Question 1. To this extent we contribute novel neural network architectures for monocular visual and visual-inertial odometry. In Chapter 5, we introduce novel self-supervised losses for learning ego-motion in the abscence of labelled data. These contributions are described in the following publications:

- Wang, S., **Clark, R.**, Wen, H., Trigoni. N. "End-to-End, Sequence-to-Sequence Probabilistic Visual Odometry through Deep Neural Networks." International Journal of Robotics Research (IJRR) Special Issue on Deep Learning in Robotics.
- Wang, S.,**Clark, R.**, Wen, H., Trigoni. N. Deep VO: Towards End-to-End Probabilistic Visual Odometry with Recurrent Convolutional Networks. To appear at IEEE International Conference on Robotics and Automation (ICRA), 2017

- **Clark, R.**, Wang, S., Wen, H., Trigoni. N., Markham, A. VINet: Visual Inertial Odometry as a Sequence to Sequence Learning Problem. Thirty-First AAAI Conference on Artificial Intelligence (AAAI), 2017

Chapter 6: The egomotion estimates from Chapter 4 provide poses with respect to some starting point. These poses drift over time and external input (e.g. GPS) is required to reduce this drift. In this chapter we consider learning to re-localize image streams in a global reference for areas that have been previously mapped. As our deep re-localization model only requires poses for training, it can be setup using any survey method available and used as a prior on global pose for the mapping system described in Chapter 7.

- **Clark, R.**, Sen, W., Wen, H., Trigoni. N. "A Deep Spatio-Temporal Model for 6-DoF Video-Clip Relocalization." IEEE International Conference on Computer Vision and Pattern Recognition, (CVPR), 2017.

Chapter 7: This chapter utilises the models proposed in Chapters 4-6 and addresses Question 2 relating to how we can integrate learning-based priors into a standard visual SLAM system for robustly reconstructing 3D maps. This work is partly based on the following publication:

- **Clark, R.**. "Dense Visual-Inertial SLAM with Deep Priors." IEEE Robotics and Automation Letters (RAL). *(submitted)*

- **Clark, R.**, Wang, S., Wen, H., Trigoni. N., Markham, A. "iSfM: Pushing the Limits of 3D Indoor Modelling using Mobile Sensing." IEEE Transactions on Mobile Computing (TMC). *(under revision)*

Chapter 8: In this chapter we address the challenge of efficently localizing in the reconstructed large-scale maps. These contributions are partly described in

- **Clark, R.**, Wen, H., Wang, S., Trigoni. N., Markham, A. Increasing the Efficiency of Visual Localization using Multi-Modal Sensing. In Proceedings of the IEEE RAS International Conference on Humanoid Robotics (Humanoids), 2016.

- **Clark, R.**, Markham A., Trigoni. N. Robust Vision-based Indoor Localization. In Proceedings of the IEEE International Conference on Information Processing in Sensor Networks (IPSN), 2015.

Chapter 3

Preliminaries

In this section we outline the basic concepts related to vision-based localization and mapping such as the image formation process and feature descriptors, as well as an overview of the auxillary sensors which we consider. As well as providing an essential background to the components of traditional localization approaches, the projection models are needed for the self-supervised losses in Chapter 5 and the hand-crafted feature descriptors form the basis of the competing traditional localizaiton methods to which we compare our proposed models and furthermore, intrinsic camera calibration and distortion is an important factor in the generalization of our proposed learning-based methods.

3.1 Sensors for Localization

3.1.1 Cameras

Most real-world cameras can be approximated relatively faithfully by a pinhole camera model. The pinhole camera model is a mathematical model that describes that describes the image formation process and allows one to relate the 3D points of the world to their 2D locations on an image. Typically, four sets of co-ordinate systems

are used in the projection process,

1. the **world coordinates** - the 3D position of a specific point in the world
2. the **camera coordinates** - the 3D position of a specific point in the camera's frame of reference (i.e. with the camera at the origin)
3. the **image coordinates** - the 3D (or homogeneous 2D position) of a point in the camera space
4. the **pixel/frame coordinates** - the 2D pixel location of a point on the camera frame

The intrinsic parameters of the camera fully parameterize the transformation from the camera coordinates to the pixel position, while the extrinsic parameters describe the transformation between the world coordinates and the camera coordinates.

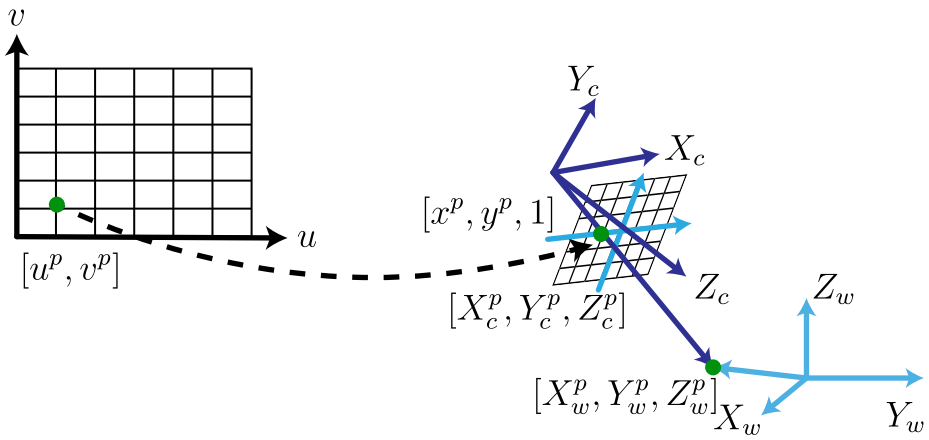


Figure 3.1: Illustration showing how a single point p is represented in the world and camera co-ordinate systems.

In the pinhole model the 3D point is mapped to the image via

$$(X, Y, Z) \rightarrow \left(\frac{f_x X}{Z} + c_x, \frac{f_y Y}{Z} + c_y \right) \tag{3.1}$$

This transformation can be represented as a matrix operation in two steps, first the camera-space coordinates are transformed to image-space using

$$\begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix} \rightarrow \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_x \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{K} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.2}$$

The matrix $\mathbf{K}$ is known as the intrinsic calibration matrix and converts camera-space coordinates to image-space. The relationships between these coordinate systems are illustrated in Fig. 3.1. To obtain the 2D pixel co-ordinates, a perspective division is performed using the Z co-ordinate of the point which results in a 2D homogeneous representation $\mathbf{p} = [x, y, 1]$ of the point on the image plane. This projection is represented using $\mathbf{p} = \pi(\mathbf{KP})$. The 2D image points can also be projected back into 3D space using the inverse projection, however, it is not possible to recover the 3D depth from a single image and thus points on the image plane are converted to rays. If the depth is available, the point can be back-projected as $\mathbf{P} = \pi^{-1}(\mathbf{KP}, z) = z\mathbf{K}^{-1}\mathbf{p}$.

In most cases, we are interested in multiple cameras where each camera has a specific position relative to each other or the world frame. The extrinsic parameters represent the position and orientation of the camera in the world frame. The extrinsic parameters can be represented in matrix notation via

$$\mathbf{T}_{CW} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

Putting these transformations in a single equation, the fully-calibrated camera

projection transformation from camera co-ordinates to pixel co-ordinates becomes

$$\lambda \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{R} \mid \mathbf{t} \end{bmatrix} \begin{bmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X^c \\ Y^c \\ Z^c \end{bmatrix}, \tag{3.4}$$

In this thesis we are primarily concerned with deriving the extrinsic calibration matrix $\mathbf{T}_{CW}$, or the pose of the camera, $\mathbf{T}^1_{CW} \dots \mathbf{T}^n_{CW}$, with respect to some fixed world-frame from a sequence of captured images.

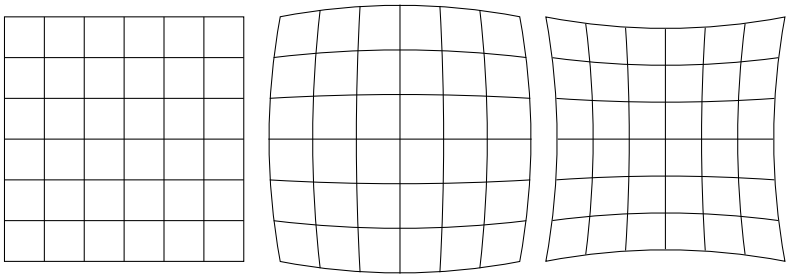


Figure 3.2: Radial distortion caused by a lens.

The above camera model only describes the protective geometry of the image-formation process and does not account for the effect of the lens. However, real cameras have lenses and lenses subject the incoming light to a range on non-linear distortions we need to be accounted for before the pinhole projection equations can be utilized. The radial and tangential models are the two most severe non-linear distortions induced by a typical lens. Both of these models can be described by the parametric equation

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = f(r, \mathbf{k}) \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.5}$$

where $[x', y'] \in \mathbb{R}^2$ are the distorted normalized pixel coordinates and $[x, y] \in \mathbb{Z}^2$ are

the undistorted normalized pixel coordinates and $r^2 = x^2 + y^2$ is the radial from the optical center. The accuracy of the undistortion can be controlled by the number of distortion parameters k_i used to

$$x_d = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{3.6}$$

$$y_d = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{3.7}$$

These distorted co-ordinates can be pre-computed and stored in a lookup-table to perform fast radial undistortion of images during runtime. The parameters describing $f(r, \mathbf{k})$ also form part of the camera's intrinsic calibration.

3.1.2 Inertial Measurement Unit

The inertial measurement unit (IMU) is a self-contained sensor that measures linear acceleration and angular velocity. IMUs also often contain a magnetometer which measures the magnetic field strength which can be used to provide information about the yaw of a platform. The most commonly used IMUs are based on microelectromechanical (MEMs) technology. Gyroscopes, and especially MEMs variants of gyroscopes, provide measurements that are subject to bias and noise. The measurement equation of a typical gyroscope is

$$\tilde{\omega} = \omega + b_{gyro} + \eta, \quad \eta \sim \mathcal{N}(0, \sigma^2_{gyro}) \tag{3.8}$$

The accelerometer measures the linear acceleration of a body along with the effect of gravity, a constant bias and measurement noise. The measurement equation of a typical accelerometer is

$$\tilde{a} = a_g + a + b_{gyro} + \eta, \quad \eta \sim \mathcal{N}(0, \sigma^2_{gyro}) \tag{3.9}$$

Table 3.1: Non-exhaustive list of parameters that need to be expertly tuned in a traditional VIO system

IMU	
a_{max}	acceleration saturation
g_{max}	gyro saturation
σ_{g_c}	gyro noise density
σ_{a_c}	accelerometer noise density
σ_{bg}	gyro bias prior
σ_{ba}	accelerometer bias prior
σ_{gw_c}	gyro drift noise density
σ_{aw_c}	accelerometer drift noise density
g	Earth's acceleration due to gravity
b_a	Accelerometer bias
T_{BS}	Body-Sensor transformation
t_d	cam-imu timestamp offset
t_r	cam readout delay

Bias and noise, however, are not the only sources of measurement inaccuracy for IMUs. The biases are, for example, subject to both temperature and time drift and thus need to be estimated and accounted for during online operation.

3.1.3 Auxillary Sensors

The camera is not the only sensor that can be used to determine the location of a mobile robot. WiFi has been a popular localization modality [49]. In this case, the robot measures the received signal strength of WiFi access points and builds a map that can be used to provide coarse localization information. Maps can also be built of the geomagnetic distortions created by building structures [54]. Although the localization accuracy of these methods is far lower than vision-based approaches, they can be used as auxillary sensors to aid the visual-localization process. In Chapter 7 of this thesis we demonstrate, we show that it is possible to use these modalities to perform efficient localization in large point cloud maps.

3.2 Feature-based visual localization

Images of a particular place or location contain a great deal of information about the scene. However, if this information is to be used for any useful purpose the raw information is inadequate as the image changes significantly under different lighting and viewing conditions. Feature descriptors thus seek to encode *distinctive* aspects of the scene which can be re-identified under a range of conditions [61].

Local feature descriptors

Local features are evaluated at many local points in an image. These points are selected with the goal of being distinctive, repetitive, invariant and present in high enough quantities in the image. Some common algorithms used for feature point detection include the Harris corner detector, FAST [37], the Difference of Gaussian detector and maximally stable extremal regions (MSER) [101]. Once feature points have been detected, a rich descriptor is extracted to describe that specific feature point in the image. The most widely used feature descriptors are:

SIFT/SURF SIFT features [96] are the highest performing and most discriminative features and are invariant to rotation, scaling and translation as well as lighting and image noise. Furthermore, they are multi-scale descriptors which allow them to be matched at varying camera distances. Each SIFT feature comprises a 128-element floating point vector. As such, SIFT features are both computationally and memory intensive. SURF features [4] were designed to overcome the performance issues at a slight decrease in matching accuracy, and can be rapidly computed using integral images. SURF features are still memory intensive, however, using 64 floating point values per descriptor.

ORB features combine an oriented FAST interest point detector with BRIEF descriptors to create a rotationally invariant binary descriptor which is fast to compute

and highly discriminative [130]. ORB features, however, are not as robust to scale changes as SURF.

Global feature descriptors

In contrast to local features, global features use a single vector to describe the entire image in a compact manner.

BoVW The Bag-of-Visual-Words (BoVW) model is the most widely used image descriptor that has been applied to a range of computer vision tasks [164]. It was originally introduced for image retrieval by Sivic and Zisserman [146]. The BoVW model represents the image as a histogram of visual word occurrences. These visual words correspond to local features (SIFT, BRISK, ORB, etc.) which have been quantized into a vocabulary during a training phase. The vocabulary is created during an offline phase by clustering a large set of descriptors extracted from a set of training images. During online operation, local descriptors are extracted and then each descriptor is quantized using the visual dictionary to form a histogram vector representing the image. The size of the vocabulary significantly affects the accuracy and computational demands of the system and provides a tunable tradeoff; with small vocabulary sizes quantization is fast yet the precision is significantly impacted and vice versa. Even with a small vocabulary, however, computing BOW histograms are too slow for real-time operation in mobile computing applications considering the high-dimensional floating point vectors that need to be quantized. With DBOW2, Galvez-Lopez [59] introduced a binary BOW approach which significantly speeds up these algorithms with little reduction in accuracy.

More descriptive coding and pooling methods to form global descriptors from local features have also been introduced in the form of VLAD [3] and Fisher vectors [123] which use the Gaussian mixture model probabilities for feature coding and a GMM adaptation procedure for pooling; sparse coding which uses a soft quantization

scheme and max pooling and uncertainty-based quantization (UNC) which uses soft quantization and histogram aggregation as in BoVW.

GIST The GIST descriptor was developed to give a holistic low dimensional representation of an image. The low dimensional vector is designed to approximate actual human perceptual dimensions of "naturalness, openness, roughness, expansion and ruggedness" [121]. Practically, this is accomplished by dividing an image into a 4×4 grid and computing orientation histograms in each square. GIST descriptors are then typically compared using L2 distance, leading to scenes with close semantic relations having short distances to one another. GIST descriptors are widely applied for web-based image retrieval [40], as well as for scene context recognition such as classifying traffic scenes [144].

3.3 Feature-based monocular structure and motion

SfM pipelines, which have been studied extensively [161, 1], allow one to accurately recover the poses of cameras and a sparse point cloud of the scene using snapshots taken from spatially varying points of view. The internal calibration of the cameras can be included in the optimization, allowing the automatic recovery of the camera calibration properties and thus requires no prior tuning or expert operating knowledge. The pipeline runs a feature detection algorithm on the images which extracts a set of distinct key-points in each image. A descriptor is extracted at each key-point, allowing key-points to be matched across images. As SfM typically operates on unstructured image collections, the matching process is carried out for all frame pairs which makes the matching very computationally expensive. Once matches have been established the cameras are registered and the tracks across frames are triangulated. Finally the entire model is bundle-adjusted to obtain the optimal structure and mo-

tion estimates. The expensive image matching along with the full bundle adjustment prohibits these pipelines from operating in real-time [134].

To achieve real-time operation, visual SLAM systems make use of numerous strategies. Firstly, they use the continuity of image streams to intelligently search for correspondences frame to frame prohibiting an exhaustive search, they perform bundle adjustment over a small local window instead of across all the images [91], thirdly they marginalize out all pose estimates apart from a select number of keyframes and perform a global optimziation (i.e. a pose-graph optimization) only over these keyframes.

3.4 Deep Learning

Recent advances in the field of machine learning have shown that it is possible to harness vast amounts of data to create robust, accurate and powerful function approximators. Deep-learning models [70, 6, 5] are a type of function approximator which during training can automatically learn to extract a powerful hierarchy of features that are most relevant or correlated with the task at hand. In so doing, they are uniquely able to operate directly on modalities such as raw images and audio which inherently contain large amounts of redundant information that traditional methods are incapable of coping with. These models have shown astounding performance in tasks where the output is of a discrete or categorical nature and fall in a bounded domain, such as object classification [72], action recognition [78], image segmentation [95] and natural language processing [26] where they have even shown the ability to learn a sequence-to-sequence mapping from one human language to another for the task of machine translation [153]. More recent work has also shown that the efficacy of deep models is not limited to learning functions with discrete output spaces but can in-fact be applied to learning any continuous regression function on input spaces

of almost arbitrarily high dimensionality and redundancy. For example, they have been used for the seemingly impossible task of regressing directly from RGB input to depth and surface normals [94], for which they achieve state-of-the-art performance [19]. They have also been used to learn functions for performing image restoration [76], denoising [77], super-resolution [38], novel view sysnthesis [80], multi-target tracking [104] and human pose prediction [156].

Most of these methods, however, focus on action spaces and output domains which are bounded, discrete and categorical and although these methods show much promise, their successful adoption for tasks such as visual ego-motion estimation and visual localization and mapping have so far eluded researchers and generated much interest in the research community [29].

Recurrent Neural Networks (RNN's) refer to a general type of neural network where the layers operate not only on the input data but also on delayed versions of the hidden layers and/or output. In this manner, the network has an internal state which it can use as a "memory" to keep track of past inputs and its corresponding decisions. RNN's, however, have the disadvantage that using standard training techniques they are unable to learn to store and operate on long-term trends in the input and thus do not provide much benefit over standard feed-forward networks. For this reason, the Long Short-Term Memory (LSTM) architecture was introduced to allow RNN's to learn longer-term trends [71]. This is accomplished through the inclusion of gating cells which allow the network to selectively store and "forget" memories.

There are numerous variations of the LSTM architecture. However, these have been shown to have similar performance on real world data [166]. The contents of the memory cell is stored in $\mathbf{c}_t$. The input gate $\mathbf{i}_t$ controls how the input enters into the contents of the memory cell for the current time-step. The forget gate, $\mathbf{f}_t$, determines when the memory cell should be emptied by producing a control signal in the range 0 to 1 which clears the memory cell as needed. Finally, the output gate $\mathbf{o}_t$ determines

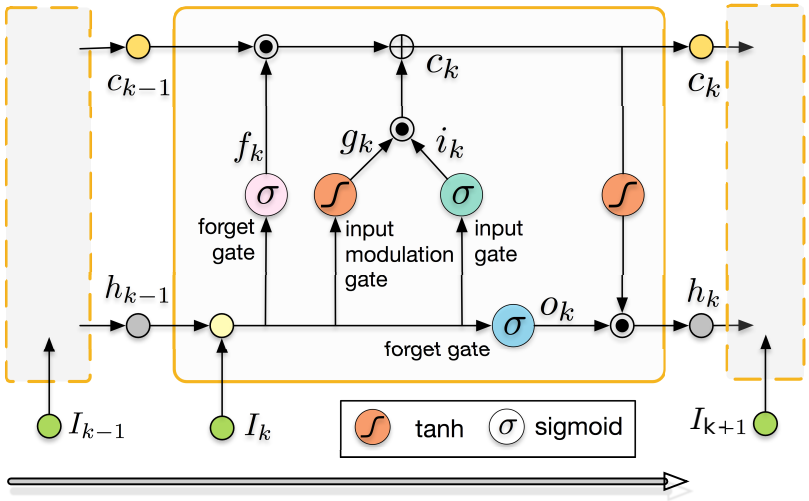


Figure 3.3: Structure of an LSTM cell.

whether the contents of the memory cell should be used at the current time-step.

The parameters, $\mathbf{W}_{i,j}, \mathbf{b}_i$ fully parameterise the operation of the network and are learned during training. The recurrent hidden layer allows the network to make use of the temporal regularity of the input data to improve its performance. This is critical for a task such as ego-motion estimation where much temporal regularity exists.

Convolutional Neural Networks (CNNs) incorporate convolutional layers in the network structure which, in contrast to the fully-connected layers used in standard RNNs and FFNs, are able to take advantage of the spatial regularity of data thereby significantly reducing the number of parameters required, allowing them to operate on high-dimensional input. In CNNs, multiple convolutional operations are applied at each convolutional layer to extract a number of features from the output map of the previous layer. The filter kernels with which the maps are convolved are learned during training. Specifically, the output of a convolutional layer is computed as [157]

Chapter 4

Learning to Estimate Ego-motion

In this chapter we propose a learning-based neural model for motion representation and fusion for the purpose of estimating the ego-motion of a moving system. Our method is fully end-to-end trainable - from input sensor data to pose output - meaning that at no stage do heuristics or manual feature engineering play a role. Compared to all previous approaches, this enables our method to naturally and elegantly distill the most relevant information from the raw input data for the task at hand. We implement, test and demonstrate that our approach not only challenges the accuracy of 30 years of research into traditional visual-inertial odometry methods but in many cases also shows major improvements in computational and memory efficiency. We show that this has numerous, often surprising, implications for the visual-inertial ego-motion estimation. Specifically, our approach is able to implicitly utilize domain specific information to significantly mitigate scale, rotational and translational drift. Furthermore, our approach does not require precise manual synchronization of the camera and IMU nor does it need explicit intrinsic calibration of the camera or extrinsic calibration between the IMU and camera. In terms of calibration, all our method requires is that the model is trained on the same camera type as is used for testing. We show that our approach is competitive with state-of-the-art traditional methods

when accurate calibration data is available and can be trained to outperform them in the presence of calibration and synchronization errors. Our method is less fragile and much more robust than existing state-of-the-art approaches.

4.1 Related publications

The publications arising from the work in this chapter include

- **Clark, R.**, Wang, S., Wen, H., Trigoni. N., Markham, A. VINet: Visual Inertial Odometry as a Sequence to Sequence Learning Problem. Thirty-First AAAI Conference on Artificial Intelligence (AAAI), 2017
- Wang, S., **Clark, R.**, Wen, H., Trigoni. N. "End-to-End, Sequence-to-Sequence Probabilistic Visual Odometry through Deep Neural Networks." International Journal of Robotics Research (IJRR) Special Issue on Deep Learning in Robotics.
- Wang, S.,**Clark, R.**, Wen, H., Trigoni. N. Deep VO: Towards End-to-End Probabilistic Visual Odometry with Recurrent Convolutional Networks. To appear at IEEE International Conference on Robotics and Automation (ICRA), 2017

4.2 Introduction

Ego-motion estimation is one of the key challenges and cornerstone requirements of machine perception. As such, solutions to these problems have been studied extensively over the past 30 years with a predominant focus on accuracy. One of the most promising approaches to achieving this goal is through the fusion of images from a monocular camera and inertial measurement unit. This setup has the advantages of being both cheap and ubiquitous and, through the complementary nature of the sensors, has the potential to provide pose estimates which are on-par in terms of accuracy

compared to more complex, heavyweight and expensive systems such as stereo and LiDAR setups. As such monocular visual-inertial odometry (VIO) approaches have received considerable attention in the robotics community [67] and current state-of-the-art approaches to VIO [51, 92] are able to achieve impressive accuracy. However, there are still only a few practical implementations of visual-inertial systems in the real-world. The main factor hindering the widespread adoption of these systems is their severe lack of robustness under changing conditions. Addressing these issues has been identified as a key requirement for future SLAM research in what has been termed the "robust perception age" [18] - underscoring the need for a fundamental shift of focus from accuracy to robustness.

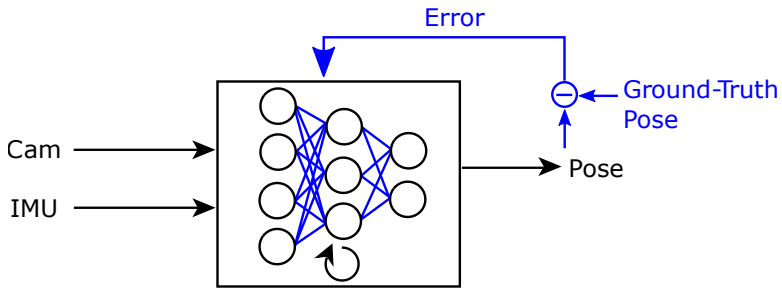


Figure 4.1: Our method robustly predicts pose from a multi-modal stream of images and IMU data.

To this extent, recent advances in the field of machine learning have shown that it is possible to harness vast amounts of data to create robust, accurate and powerful function approximators. Deep-learning models [70, 6, 5] are a type of function approximator which during training can automatically learn to extract a powerful hierarchy of features that are most relevant or correlated with the task at hand. In so doing, they are uniquely able to operate directly on modalities such as raw images and audio which inherently contain large amounts of redundant information that traditional methods are incapable of coping with. These models have shown astounding performance in tasks where the output is of a discrete or categorical nature and fall in a bounded domain, such as object classification [72], action recognition [78], image

segmentation [95] and natural language processing [26] where they have even shown the ability to learn a sequence-to-sequence mapping from one human language to another for the task of machine translation [153]. More recent work has also shown that the efficacy of deep models is not limited to learning functions with discrete output spaces but can in-fact be applied to learning any continuous regression function on input spaces of almost arbitrarily high dimensionality and redundancy. For example, they have been used for the seemingly impossible task of regressing directly from RGB input to depth and surface normals [94], for which they achieve state-of-the-art performance [19]. They have also been used to learn functions for performing image restoration [76], denoising [77], super-resolution [38], novel view sysnthesis [80], multi-target tracking [104] and human pose prediction [156].

Although these methods show much promise, their successful adoption for tasks such as visual-odometry and sensor fusion have so far eluded researchers. Effectively harnessing these data-driven methods for motion estimation and mapping has the potential to revolutionise the field and realise the long-standing [18] goal of machine perception - robust, lifelong SLAM.

Inspired by the success of deeplearning models for processing raw, high dimensional data, we propose in this chapter a new paradigm for visual-inertial sensor fusion by regarding it as a sequence-to-sequence regression problem. The resulting approach is a fully trainable end-to-end model for performing visual-inertial odometry. Our contributions are as follows

- We present the first system to perform visual-inertial odometry that is end-to-end trainable.

- We develop a recurrent network architecture, equipped with convolutional layers that allows us to homogeneously process visual and inertial data in the pose prediction process. Our architecture generates a common feature representation

for both the visual and the inertial data, reformulating the fusion process as a simple vector concatenation operation

- We propose to parametrise the frame-to-frame motion as a 6D element of the Lie algebra $se(3)$ which is up-graded to a homogeneous transform in the Lie group $SE(3)$ using a novel pose concatenation layer
- We consider a probabilistic variant of this pose parametrisation where we learn a maximum-likelihood covariance model for the pose predictions and propagate them using a differentiable information filter formulation.
- We further propose a novel training method in which we jointly train the frame-to-frame and global pose objectives
- Finally, we evaluate our method and then demonstrate its advantages over traditional methods on real-world data, including the KITTI and EuRoC datasets.

The rest of this chapter is organized as follows: in Section 4.3 we outline the related work and traditional state-of-the-art approaches to visual inertial odometry as well as existing learning-based methods. We then provide some background on the neural network models we use in our work and in Section 4.5 we describe our novel network architecture. The results on the selected open datasets are presented in Section 7.6.

4.3 Related work

In general, the vast taxonomy of approaches can be classified as direct, indirect, sparse or dense. The direct vs. indirect classification refers to the type of observations used for motion estimation, while dense vs. sparse refers to the nature of the depth or geometry estimation. Direct approaches use as observations raw pixel

intensities while indirect approaches use keypoint locations. Indirect approaches such as PTAM [84] and ORB-SLAM [112] are generally faster, more robust to noise and inaccurate initialization but can only make use of limited information in each frame (i.e. at detected keypoints). Direct approaches such as LSD-SLAM [44], SVO [53] and DSO [43] can make use of the entire image but require good initialization and higher frame-rates to prevent tracking failures. Dense approaches like DTAM [116] estimate the depth of the entire image using strong, smooth priors and optimization of the photometric error, while sparse approaches like monoSLAM [33] independently estimate the depth of individual key-point features using triangulation. Dense approaches are more robust in feature-less areas but are very computationally intensive and require high frame-rates, while sparse approaches are prone to feature depletion.

All visual odometry algorithms exhibit considerable drift over time. Visual simultaneous localization and mapping (vSLAM) algorithms thus build an on-line visual map of the world in addition to estimating the camera ego-motion. vSLAM systems build on visual odometry approaches by incorporating a "place" recognition component to detect loop closures and re-localize when the system detects that it is lost. Loop-closure methods recognise previously visited places, "closing the loop" of poses and distributing accumulated errors across the entire path. A number of competing state-of-the-art vSLAM systems exist. ORB-SLAM [112] is a feature-based SLAM method based on ORB features. By establishing map point correspondences using the ORB descriptor, ORB-SLAM is fairly robust to occlusions and fast motions. It locally optimizes the keyframe poses and map points through a bundle adjustment procedure and detects loop closures for appearance-based place recognition. SVDO [46] is a (semi) dense method for visual pose estimation from a monocular camera. It uses a method similar to that employed by RGB-D slam systems. It first estimates a depth map using a sequence of frames and then aligns subsequent depth maps to form a coherent model. In this chapter, we focus on ego-motion estimation in an odometry

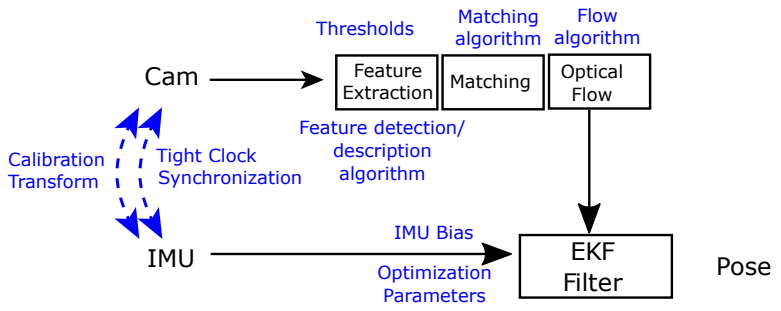


Figure 4.2: Standard visual-inertial odometry pipeline. Each component of the pipeline needs extensive hand-tuning before operation.

framework and do not consider loop-closure detection and map-building processes.

Traditional monocular solutions are unable to observe the scale of the scene and will always be subject to both scale drift and a scale ambiguity. To correct scale drift and recover metric scale, additional information needs to be introduced. This can be accomplished through a constrained VO solution. Constrained VO/SLAM methods such as [147] introduce additional information to constrain the solution and recover metric scale. The additional information can take the form of structure or motion constraints. Structural constraints involve detecting and identifying structures in the scene which have a known size or dimension - for example, many traditional solutions use the height of the camera from the ground-plane to recover and correct scale. More recent methods combine object recognition with the VO [22, 124, 133, 147] to produce a map of the objects along with a scale-corrected trajectory.

Fusing the visual data with an additional sensing modality, such as an inertial measurement unit (IMU), is perhaps the most promising means of obtaining low-cost, accurate and robust tracking without relying on scene appearance. In theory, the complementary nature between inertial measurements from an IMU and visual data should enable highly accurate ego-motion estimation under any circumstances. For example, vision-only tracking is not possible when a platform undergoes abrupt, frantic movements as the large inter-frame motion prevents any visual overlap between

frames, thus making it physically impossible to estimate motion using the visual data alone. This is often the case in VR/AR applications in indoor environments where these techniques are plagued by intermittent occlusions. The IMU can easily allow a system to track through these visual occlusions by integrating the acceleration and angular velocity data, providing high-rate position and orientation estimates when the visual information is unavailable. However, the pose estimates derived from the IMU drift rapidly due to error compounding associated with the integration process. In state-of-the-art systems, drift is mitigated using tightly-coupled fusion of the visual and inertial data through the use of either a filter-based or optimization-based procedure. Filter-based methods such as the Multi-state Constraint Kalman Filter (MSCKF) [110], although being more robust, consistently under-perform their optimization-based counterparts in terms of accuracy.

4.3.1 Parameters

Successfully calibrating a traditional VIO system requires tuning a large number of parameters. A non-exhaustive list of the typical calibration parameters are shown in Tbl. 3.1. These parameters relate to the camera itself, the IMU device, the camera-IMU combination, as well as the processing parameters of the vision algorithm. Although these parameters can be obtained on a system-by-system case, three factors complicate large deployments. These factors are 1) the non-deterministic variability of the parameters between individual deployments 2) the non-deterministic variability of the parameters between time-steps and 3) the drift of the parameters over time (for example as the system heats up). Numerical values for the camera-IMU time-offset and camera readout time of an iPhone and iPad device are shown in Tbl. 4.1 [141].

As is evident, the time-delay and readout time variability is in the order of 10's of

Table 4.1: Camera-IMU time-offset variability for the iPhone and iPad devices [141]

	$t_d(\sigma)$	$t_r(\sigma)$
iPhone	0.0114s (0.0038s)	-0.0224s (0.0001s)
iPad	0.0220s (0.00036s)	-0.0213s (0.00017s)

ms. This, along with the multitude of other parameters that need to be calibrated on a per-device basis, make large scale deployments of visual-inertial systems very difficult. End-to-end trainable approaches are a very promising means of overcoming these issues of robustness. Not only can all the parameters of the system be automatically determined during training, such approaches could possibly learn to become robust to the inherent variability of the parameters.

4.3.2 Deep-learning for Ego-motion Estimation

Some deep-learning approaches have been proposed for visual odometry, however, a neural network approach has never been used in any form for end-to-end fusion of visual-inertial data. In [86], a stereo-VO method is presented where they extract motion by detecting "synchronicity" across the stereo frames. [30] investigated the feasibility of using a CNN to extract ego-motion from pre-processed optical flow frames. In [35] the feasibility of using a CNN for extracting the homography relationship between frame pairs was shown. While these approaches have relied on the availibility of ground-truth data, there has been some work on using self-supervised losses for training. Garg et al. [60] introduced a self-supervised loss for training monocular depth prediction networks using calibrated stereo pairs. Godard et al. [65] proposed a similar approach, but used a loss function based on the consistency of projections in the left and right frames of stereo pairs. Zhou et al. [170] introduced a reprojection-error based loss for simultaneously training depth and motion prediction networks. Although they do not evaluate the performance of their networks when trained using labelled data, their network architecture is the closest existing work to ours; being the

only other model that takes as input multiple images and predicts a series of poses as output. However, instead of using a single CNN to predict multiple different poses, our method aims to make better use of temporal information. In particular, we use a CNN, replicated across multiple time steps and embedded in a recurrent formulation to model temporal dependencies.

To the best of our knowledge, the only application of deep-learning methods to visual-inertial data is [125] where the IMU data is processed using a recurrent neural network and fused with the output of a marker-based visual tracking system using an extended Kalman filter.

4.4 Background

In this section we describe the standard components on which our network is based i.e. a combination of recurrent and convolutional neural network layers.

4.5 Our Approach

In this section we describe our complete model and the novel components which we introduce.

Our sequence-to-sequence regression-based approach to visual-inertial odometry, is shown in Fig. 4.4. The model consists of an CNN-RNN network which has been tailored to the task of visual-inertial odometry estimation. The entire network is differentiable and thus trainable end-to-end for the purpose of ego-motion estimation. The input to the network is monocular RGB images and IMU data which is a 6 dimensional vector containing the x, y, z components of acceleration and angular velocity measured using a gyroscope. The output of the network is a 7 dimensional vector - a 3 dimensional translation and 4 dimensional orientation quaternion - representing the change in pose of the robot from the start of the sequence. In essence, our network

learns the following mapping which transforms input sequences of images and IMU data to poses

$$\text{VIO} : \left\{\left(\mathcal{R}^{W\times H}, \mathcal{R}^{6}\right)_{1:N}\right\} \rightarrow \left\{\left(\mathcal{R}^{7}\right)_{1:N}\right\} \tag{4.1}$$

Our method processes the multi-modal data in two streams, each stream producing a feature vector which is fused through a concatenation operation. The IMU feature vector is generated by the mapping

$$F_{IMU} : \left\{\left(\mathcal{R}^{6}\right)_{1:N}\right\} \rightarrow \left\{\left(\mathcal{R}^{100}\right)_{1:N}\right\} \tag{4.2}$$

where $D_{F_{IMU}}$ is the dimension of the IMU feature vector. Similarly, the image feature vector is generated by

$$F_{IMG} : \left\{\left(\mathcal{R}^{W\times H}, \mathcal{R}^{W\times H}\right)_{1:N}\right\} \rightarrow \left\{\left(\mathcal{R}^{100}\right)_{1:N}\right\} \tag{4.3}$$

These mappings are parameterised using a deep convolution neural network with recurrent layers. We now describe the detailed structure of our model which integrates the following components.

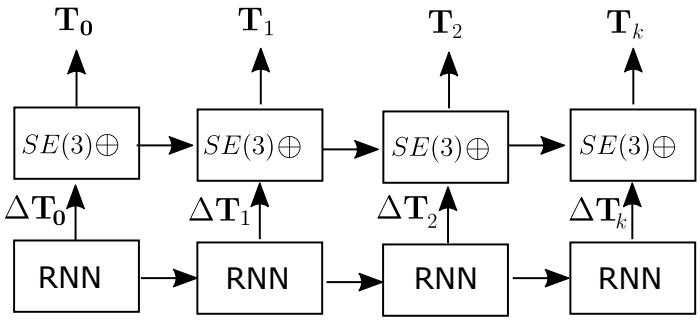


Figure 4.3: Integration of transformations between frames on SE(3).

4.5.1 SE(3) Composition of Transformations

The pose of a camera relative to an initial starting point is conventionally represented as an element of the special Euclidean group $SE(3)$ of transformations. Elements of $SE(3)$ are transformation matrices which consist of a rotation from the special orthogonal group $SO(3)$ and a translation vector,

$$\mathbf{T} = \left\{ \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} \;\middle|\; R \in SO3 \;,\; T \in \mathcal{R}^3 \right\}. \tag{4.4}$$

Producing transformation estimates belonging to $SE(3)$ is not straightforward as the $SO(3)$ component needs to be an orthogonal matrix. The Lie Algebra $se(3)$ of $SE(3)$, however,

$$\frac{\xi}{dt} = \left\{ \begin{pmatrix} [\omega]_\times & v \\ 0 & 0 \end{pmatrix} \;\middle|\; v \in so(3) \;,\; v \in \mathcal{R}^3 \right\}, \tag{4.5}$$

can be described by components which are not subject to orthogonality constraints. Conversion between $se(3)$ and $SE(3)$ is then easily accomplished using the exponential map

$$\exp:\ se(3) \rightarrow SE(3) \tag{4.6}$$

In our network, a CNN processes the monocular sequence of images to produce an estimate of the frame-to-frame motion undergone by the camera. The CNN thus performs the mapping from two images to the lie algebra $se(3)$. We then convert these to the special euclidean group $SE(3)$ where the individual motions can then be composed in $SE(3)$ to form a trajectory. In this manner, the function that the CNN needs to approximate remains bounded over time as the frame-to-frame motion undergone by the camera stays in a well-defined range over the course of the trajectory. In addition, these frame-to-frame motions are usually temporally consistent on

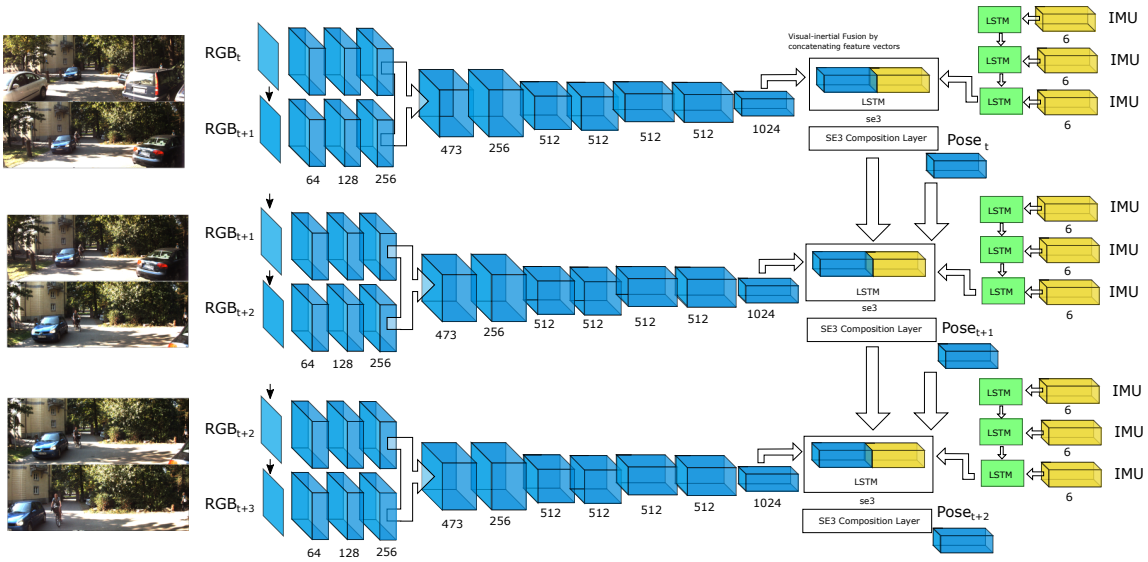


Figure 4.4: The proposed architecture for visual-inertial odometry. The network consists of a core LSTM processing the pose output at camera-rate and an IMU LSTM processing data at the IMU rate.

natural platforms - a platform travelling at a constant velocity usually continues to do so until it comes to an abrupt stop etc. We leverage this fact by feeding in the output of the CNN to a RNN which can learn the natural patterns of the platform.

4.5.2 Multi-rate LSTM

In the problem of visual-inertial odometry we are faced with the challenge of the data streams being multi-rate i.e. the IMU data often arrives at an order of magnitude (typically 10×) faster (100 Hz) than the visual data (10 Hz). To accommodate for this in our proposed network, we process the IMU data using a small LSTM at the IMU rate. We propogate the hidden state at faster rate and the final hidden-layer activation of the IMU-LSTM is carried over to the odometry LSTM which fuses the visual and inertial representations to produce a pose estimate.

4.5.3 Optical Flow Initialization

The CNN takes two sequential images as input and, similar to the IMU LSTM, produces a single feature-vector describing the motion that the device underwent during the passing of the two frames which is used as input to the Core LSTM. We initially experimented with two frames fed directly into a CNN pre-trained on the imagenet dataset, however, this showed incredibly slow training convergence and disappointing test performance. We therefore used as our base a network trained to predict optical flow from RGB images [50]. Our CNN mimics the structure of Flownet [50] up to Conv6 where we removed the layers which produce the high-resolution optical flow output and feed in only a $1024 \times 6 \times 20$ which we flatten and concatenate with the feature vector produced by the IMU-LSTM before being fed to the Core LSTM. We initialize this part of the model with the origin weights from Flownet [50], and fine-tune the entire network on our dataset.

4.5.4 Explicit State feedback

In the traditional LSTM model, the hidden state is carried over to the next time-step, but the output itself is not fed back to the input. In the case of odometry estimation the availability of the previous state is particularly important as the output is essentially an integration of incremental displacements at each step. Thus for our LSTM, we directly connect the output pose produced by the $SE(3)$ integration layer, back as input to the Core LSTM as illustrated in Fig. 4.5.

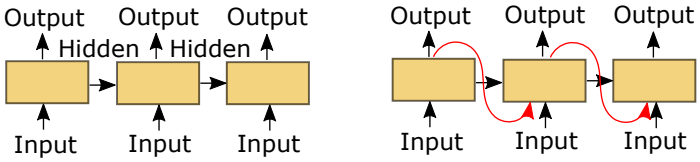


Figure 4.5: Explicit state feedback used in the LSTM.

4.5.5 Uncertainty Prediction

Odometry is inevitably subject to drift and therefore for the purpose of navigation are used in tandem with loop-closure, map-matching or pose-graph optimization methods to create SLAM systems. A key aspect of integrating odometry measurements into these larger systems is the availability of an estimate of the odometric uncertainty or the covariance of the pose estimate. Much effort in the design of traditional VIO approaches has gone into obtaining a reliable estimate of this covariance. In these traditional approaches, the output uncertainty is determined using sensor noise models. A number of works exist which attempt to estimate output uncertainty in deep-leaarning models. One such work is that of Gal et al. [58] which uses drop-out as a means of determining the model's uncertainty in representing the data.

However, the neural network based approach gives us a simple and effective means of obtaining multi-modal uncertainty estimates through the use of the mixture density networks (MDN) approach [10]. In the MDN the final layer takes as input the hidden layer of the network and adds a linear fully-connected layer which produces an output that has the same dimensionality as the parameterised pose proposal distribution, i.e.

$$y_t = \left[\{ \tilde{\phi}_i, \tilde{\mu}_\mathbf{i}, \tilde{\mathbf{\Sigma}}_\mathbf{i} \}_{i=1}^{K} \right] \tag{4.7}$$

where the first K outputs parametrise the multi-modal Gaussian mixture model corresponding to the position proposal,

$$p(\mathbf{x_{t+1}}|\mathbf{x_t}) = \sum_{i=1}^{K} \tilde{\phi}_i \mathcal{N}(\tilde{\mu}_\mathbf{i}, \tilde{\mathbf{\Sigma}}_\mathbf{i}) \tag{4.8}$$

The likelihood of the outputs is used as the loss function for training the network. The likelihood is computed from the network outputs and the output of the visual odometry as follows

$$\mathcal{L}(w, y_t) = -\log p(\mathbf{x_{t+1}}|\mathbf{x_t}, \omega_\mathbf{t}, \mathbf{m_t}) \tag{4.9}$$

where $\mathbf{x}_w(t)$ is the "correct" pose. The derivatives of all the cells can then be easily computed.

Choosing the number of components, K, is an important consideration. When $K = 1$, we see that Eqn. 4.8 reduces to a single Gaussian. As training with a large number of components in the mixture model is difficult due to mode collapse, we use choose $K = 1$ for most of our experiments. This is sufficient for out purposes as a single component is sufficient to provide a reasonable estimate of uncertainty for the output pose. This uncertainty is important for Chapter 6 where we fuse the outputs of our network with a posegraph optimization.

4.5.6 Computational requirements

The computational requirements needed for odometry prediction and the storage space required for the model are directly affected by the number of parameters used to define the model. For our network in Fig. 4.4, the parameters are the weight matrices of the LSTM for both the IMU LSTM and the Core LSTM as well as the CNN network which process the images. For our network we use LSTMs with 2 layers with cells of 2000 units. Our CNN total of $55M$ trainable weights. A forward pass of the network takes on average 10ms on a single TitanX, making it feasible to run in real time at 100 Hz.

4.6 Training

In this section we describe the training process.

4.6.1 Optimization

The entire network is trained using Backpropagation Through Time (BPTT). We use standard BPTT which works by unfolding the network for a selected number

Algorithm 1 Joint training of $se(3)$ and $SE(3)$ loss

while $i \leq n_{iter}$ **do**

$w^{1:n} = w^{1:n} - \lambda_1 \frac{\partial \mathcal{L}_{SE(3)}(w^l, x_t)}{\partial w^l}$

$w^{1:j} = w^{1:j} - \lambda_2 \frac{\partial \mathcal{L}_{se(3)}(w^l, x_t)}{\partial w^l}$

end while

Algorithm 1: The joint-loss training method.

of timesteps, T, and then applying the standard backpropagation learning method involving two passes- a forward pass and backward pass. In the forward pass of BPTT, the activations of the network are calculated successively for each timestep from time $t = 1$ to T. Using the resulting activations, the backward pass proceeds from time $t = T$ to $t = 1$ calculating the derivatives of each output unit with respect to the layer input (x^l) and weights of the layer (w^l). The final derivatives are then determined by summing over the time-steps. Stochastic Gradient Decent (SGD) with an RMSProp adaptive learning rate is used to update the weights of the networks determined by the BPTT. SGD is a simple and popular method that performs very well for training a variety of machine learning models using large datasets [12]. Using SGD, the weights of the network are updated as follows

$$w^l = w^l - \lambda \frac{\partial \mathcal{L}(w^l, x_t)}{\partial w^l} \tag{4.10}$$

The learning rate (λ), which determines how strongly the derivatives influence the weight updates during each iteration of SGD. For all our training we select the best learning rate.

Training long, continuous sequences with the high dimensional images as input requires an excessive amount of memory. To reduce the memory required, but still keep continuity during training, we use the training structure illustrated in Fig. 4.6 where the training is carried out over a sliding window of batches, with the hidden state of the LSTM carried over between windows.

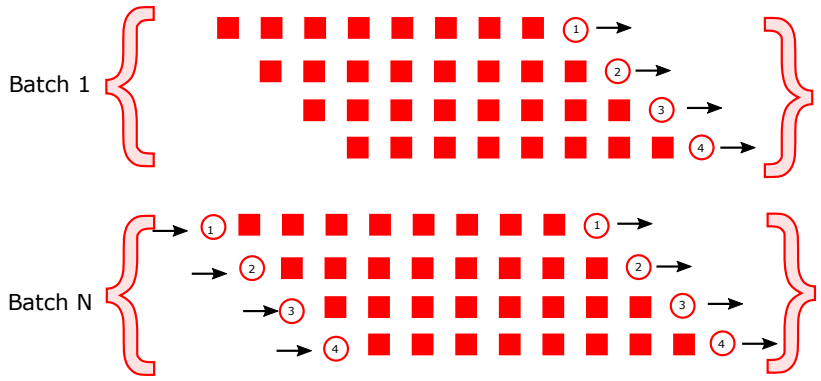


Figure 4.6: Batch structure used for training on long odometry sequences. The figure depicts a setip where each batch consists of 4 sequences and each sequence has 8 timesteps. The hidden state of the RNN for each sequence in the batch is carried over to the next batch, allowing long, continuous sequences to be trained.

Finally, we found that training the network directly through the $SE(3)$ integration is particularly difficult as the training procedure suffers from many local minima. In order to overcome this difficulty, we propose the joint training method shown in Alg. 1 which alternatively backpropogates the error of the $se(3)$ frame-to-frame predictions and the $SE(3)$ full pose relative to the start of the sequence, with the losses computed from the frame-to-frame pose as $\mathcal{L}_{se(3)} = \alpha \sum ||\omega - \hat{\omega}|| + \beta||v - \hat{v}||$. For full pose in $SE(3)$, we use a quaternionic representation $\mathcal{L}_{SE(3)} = \alpha \sum ||\mathbf{q} - \hat{\mathbf{q}}|| + \beta||T - \hat{T}||$. These are trained jointly as in Alg. **??**.

4.7 Experiments

In this section we present results evaluating the proposed method in terms of accuracy and robustness to calibration and synchronization errors and provide comparisons to traditional methods. We implemented our model using the Theano library [7] and carried out all our training on a Titan X.

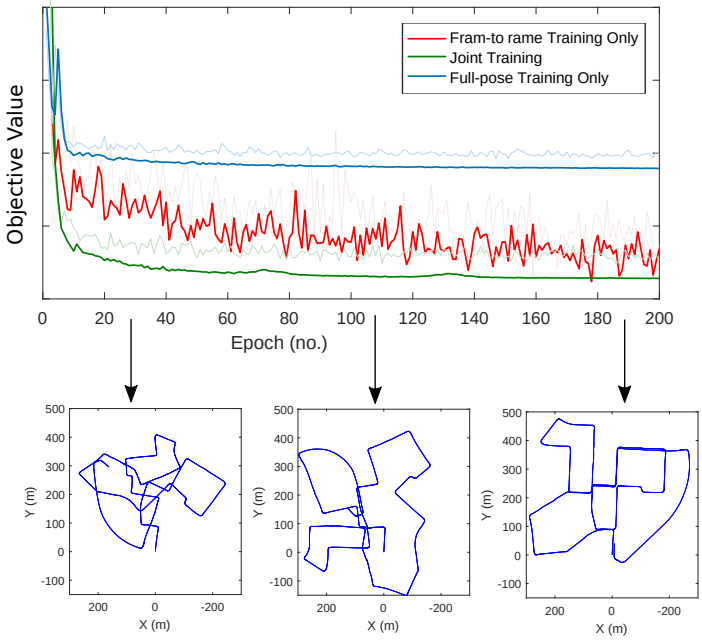


Figure 4.7: Training performance of the network using the $SE3$ layer. The visualized trajectory corresponds to the joint training.

4.7.1 Competing approaches

Here we briefly describe the competing state-of-the-art monocular visual-inertial approaches against which we compare.

ROVIO: The Robust Visual-Inertial Odometery algorithm (ROVIO) [11] is a filter-based monocular visual-inertial odometry method. In order to achieve a high level of robustness ROVIO employs a direct tracking approach and on-line estimation of the IMU-Camera calibration parameters. The integration of the visual and inertial data is achieved through a tightly-coupled formulation in the form of an extended Kalman filter.

OK-VIS: Open Keyframe-based Visual-Inertial SLAM (OK-VIS) [92] is an optimization based visual-inertial odometry approach. OK-VIS integrates IMU and visual reprojection error into a single non-linear cost function. Optimization is carried out over a bounded size window by marginalizing out over sets of frames and keeping only landmarks that are present in key-frames. OK-VIS estimates the IMU bias online,

but not the camera-IMU calibration which has to be specified beforehand.

SWF: The ROVIO and OK-VIS methods do not perform well on the KITTI dataset due to the low frame-rate and lack of precise timing synchronization. For comparison on this dataset, we thus use the sliding window filter (SWF) approach outlined in [73].

4.7.2 Datasets

In this section we describe the 3 challenging standard datasets that we use to test the robustness and accuracy of our approach.

UAV: EuRoC Dataset We first evaluate our approach on the publicly-available indoor EuRoC micro-aerial-vehicle (MAV) dataset [16]. The data for this dataset was captured using a AscTec Firefly MAV with a front-facing visual-inertial sensor unit with tight synchronization between the camera and IMU timestamps. The images were captured by a global-shutter camera at a rate of 20 Hz, and the acceleration and angular rate measurements from the IMU at 200 Hz. The 6-D ground-truth pose was captured using a Vicon motion capture system at 100 Hz. In order to provide an objective comparison to the closest related method in the literature, the optimization-based Open Keyframe VISual-inertial odometry (OKVIS) [92] method is used for comparison. As we are interested in evaluating the odometric performance of the methods, no loop-closures are performed. For testing we select a specific sequence and then train on the other remaining sequences.

Autonomous Driving: KITTI We further test the performance of our proposed method using the KITTI odometry benchmark [63, 62]. The KITTI dataset comprises of 11 sequences collected from atop a passenger vehicle driving around a residential area with accurate ground-truth obtained from a Velodyne laser scanner and GPS unit. We use sequence 02,03,04,05,06,07,09,10 for training and 00,01,08 for testing. The monocular images and ground-truth are sampled at 10 Hz, while the IMU data

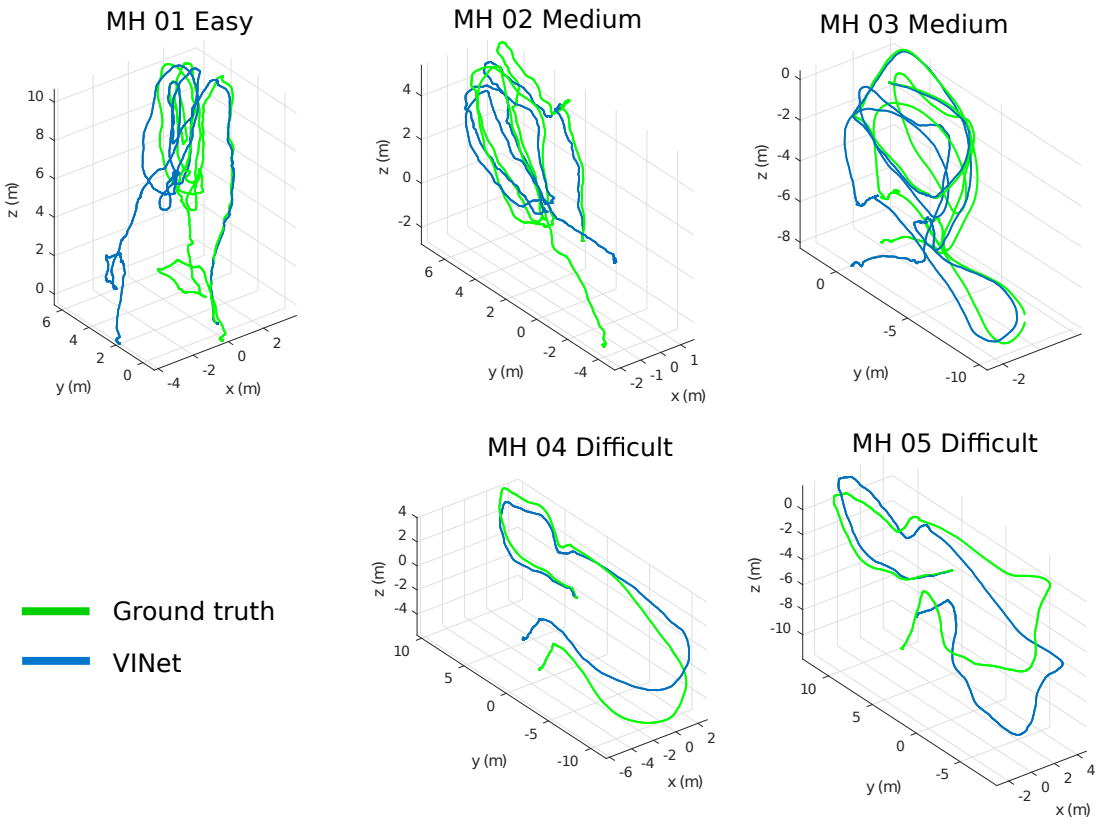


Figure 4.8: The EuRoC trajectories.

is recorded at 100 Hz. Calbration between the camera and LiDAR (not used in this thesis) was carried out using a reed switch which triggered the front facing camera on each LiDAR scan [62]. The camera and IMU, however, cannot be synchronized in this manner and instead the closest LiDAR timestamp is associated with the IMU readings [62]. The KITTI dataset therefore has a 5ms variability in the time-offset between the IMU and camera. For this reason, it is impossible to successfully run OKVIS [92] on this dataset. For comparison, we only compare to the SWF approach [73]. We follow the standard KITTI evaluation metrics where we calculate the error for $100m, 200m, \ldots, 500m$ sequences. Being recorded outdoors on structured roads, this dataset exhibits negligible motion blur and the trajectories follow very regular paths. However, it has different characteristics compared to the indoor EuRoC dataset

which still make it very challenging. For example, the vehicle path contains many sharp turns and cluttered foliage areas which make data association between frames difficult for traditional visual odometry approaches.

4.8 Learning with Self-supervised Losses

For the results in the previous section we used labelled ground-truth poses to provide the learning signal for training our odometry networks. However, in many cases the equipment or setup required to generate these ground-truth poses may not always be available. In this section we show that it is possible to the ego-motion prediction networks from unlabelled data through a self-supervised learning signal. To this extent we propose two self-supervised losses; the first relates to the visual sensor only, is based on the photometric error and has the desirable property of allowing us to also train the optical flow part of the network from scratch. The second loss that we proposed is related to the combination of visual and inertial sensors and is based on finding a feature representation that best aligns these two sensors.

The optical flow of a pixel $p = (x, y)$ is related to the incremental relative motion of the platform.

$$\dot{p} = \frac{1}{Z} \underbrace{\begin{bmatrix} xV_z - V_x \\ yV_z - V_y \end{bmatrix}}_{\text{translation flow}} + \underbrace{\begin{bmatrix} xy & -(1+x^2) & y \\ (1+y)^2 & -xy & -x \end{bmatrix}}_{\text{rotation flow}} \Omega \tag{4.11}$$

The flow induced by the rotational motion of the platform directly relates the rotation of the platform, the pixel's coordinate in the image and the flow at that pixel and thus it is easy to set up a cost function for training a network to predict rotations. However, it is evident that the translational part of the equation depends on another quantity i.e. the pixel's depth. Therefore, in order to obtain a self-

supervised loss for the 6-DoF motion we introduce a separate network in addition to the motion prediction CNN described in the previous chapters. This network takes as input the RGB image and predicts the depth values $Z : \mathbb{R}^2 \rightarrow \mathbb{R}$ corresponding to each pixel.

To train the network on unlabelled image sequences we learn weights that minimize the photometric error between consecutive frames of the video assuming that brightness constancy is maintained. To align the pixels for error computation, we warp the image using bi-linear interpolation.

$$I'(x_i^t, y_i^t)_i^c = \sum_n^H \sum_m^W I^c(n, m) \max(0, 1 - |x_i^s - m|) \max(0, 1 - |y_i^s - n|) \tag{4.12}$$

Using the sampled co-ordinates, the total photometric loss value is computed as the mean of the differences across the training batches

$$\mathbb{L}^{photo} = \frac{1}{nwhc} \sum_{x,y} ||I_t(x, y) - I_{t+1}(x', y')|| \tag{4.13}$$

4.9 Motion Correlations with Deep CCA

Canonical correlation analysis (CCA) enables the unsupervised learning of feature representations of multiple modalities which are maximally correlated. These features can then easily be fused and used for prediction tasks. Deep CCA, proposed in [2], employs a deep network to perform the feature extraction. Deep CCA has been used to learn multi-lingual word embeddings [98], to learn features for face recognition that are invariant to pose [36] and for matching images and text [163]. The features learned by Deep CCA are maximally correlated and thus ideally contain only information that is consistent across both modalities - such as motion. In this work we thus

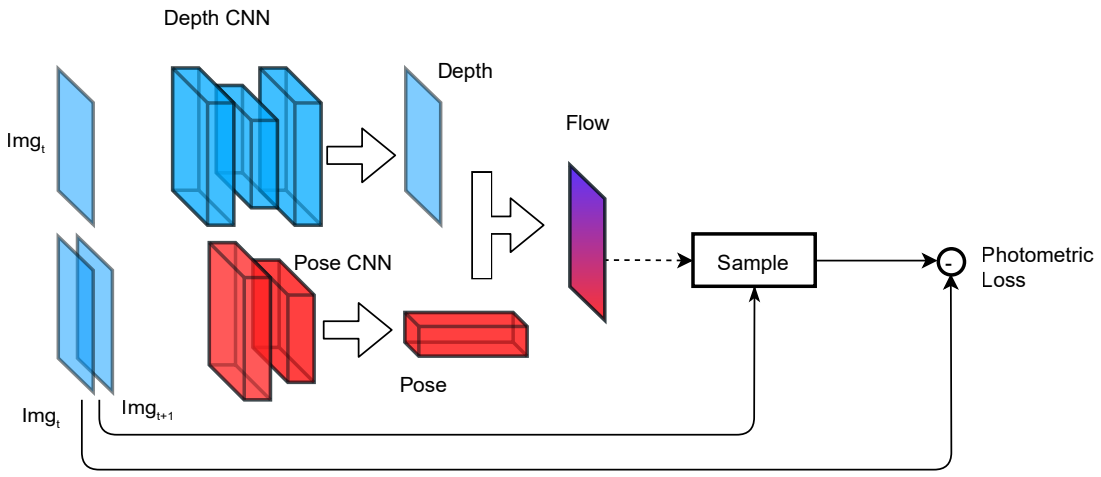


Figure 4.9: Overview of the per-frame CNN architecture for the self-supervised photometric loss. The figure shows a single timestep of the CNN for pose and depth prediction. These networks are repeated for each subsequent frame as in Figure 4.4.

investigate the use of a Deep CCA approach for performing unsupervised learning of motion features for visual-inertial data.

For performing unsupervised training of the network we optimize the Deep CCA objective [2]

$$J(\theta_1, \theta_2) = \text{corr}(f_1(X_1; \theta_1), f_2(X_2; \theta_2)) \tag{4.14}$$

where θ_1 and θ_2 are the combined parameters $(\mathbf{W_j^l}, \mathbf{b}_j^l)$ of the visual and inertial networks, respectively. The correlation function is computed using the sum of the singular values of the matrix, M, computed from the centered covariance $\Sigma_{11}\Sigma_{12}$ and cross-covariance Σ_{12} matrices. Writing the output of f_1 and f_2 as F_1 and F_2, the corrrelation objective is computed

$$\text{corr}(F_1, F_2) = tr(M'M)^{\frac{1}{2}} \tag{4.15}$$

where

$$M = \Sigma_{11}^{-\frac{1}{2}}\Sigma_{12}^{-\frac{1}{2}}\Sigma_{22}^{-\frac{1}{2}} \tag{4.16}$$

For training, the gradients of the correlation function computed using the SVD of $M = UDV'$ with

$$\frac{\partial corr(F_1, F_2)}{\partial F_1} = \frac{1}{n-1}\left(2\Delta_{11}F_1 + \Delta_{12}F_2\right) \tag{4.17}$$

with

$$\Delta_{11} = \Sigma_{11}^{-\frac{1}{2}} UV' \Sigma_{22}^{-\frac{1}{2}} \tag{4.18}$$

$$\Delta_{12} = \Sigma_{11}^{-\frac{1}{2}} UDU' \Sigma_{11}^{-\frac{1}{2}} \tag{4.19}$$

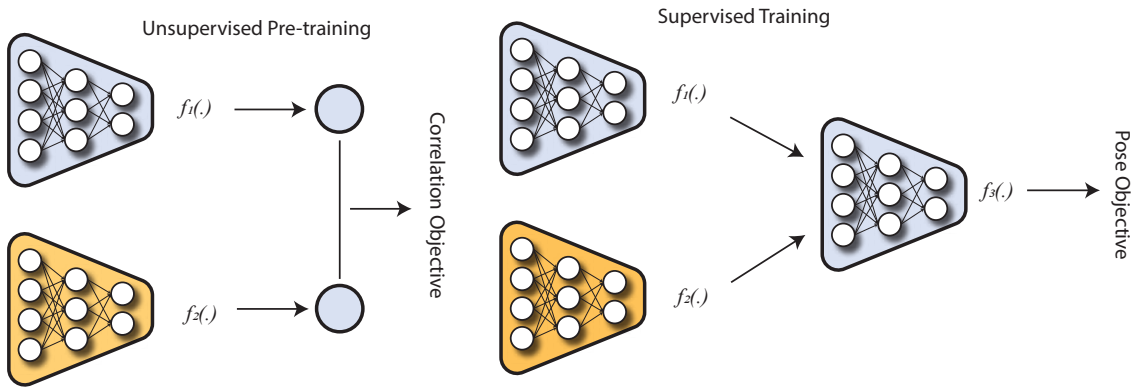
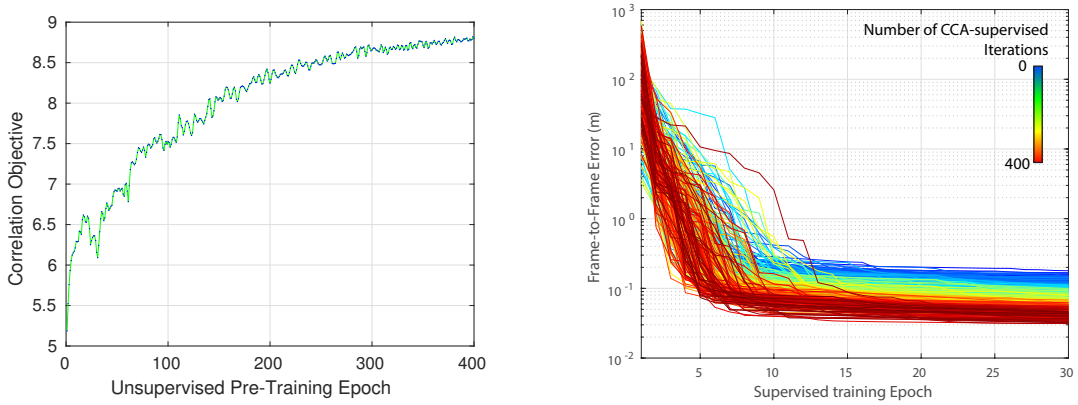


Figure 4.10: Illustration of the use of the CCA-based motion representation method during unsupervised pre-training and during supervised training [2].

In our case, the feature representations, F_1 and F_2 for which the correlation is maximized corresponds to the output of the visual CNN and the inertial LSTM, before the concatenation in the core LSTM.

4.10 Results

Here we evaluate the performance of the self-supervised photometric loss and the semi-supervised CCA objective on the task of ego-motion prediction as presented in the preceeding chapter as a supervised learning problem. First, we evaluate the CCA



(a) The CCA objective over 400 epochs of unsupervised pre-training.

(b) Effect of unsupervised motion-representation learning using CCA.

Figure 4.11: Performance of the correlation-based pretraining method. The figure shows the correlation objective over 400 epochs and the corresponding effect that this has on the supervised training (red- 400 epochs CCA training, blue - 0 epochs CCA pretraining).

objective by training the selected KITTI sequences on 400 epochs of unsupervised training using the visual-inertial data. We plot the resulting correlation value (given by Eqn. 4.17) in Figure 4.11a. Figure 4.11a shows that the SGD-based training successfully increases raises the correlation of the learned feature projections from the visual-inertial sensors by $\approx 78\%$ and suggests that the CCA training is helping the network to learn a representation that is meaninful for the ego-motion regression task - as the effect of ego-motion is experienced by both sensors simultaneously, albeit through a completely unknown but constant transformation to which the learned featurespace should be invariant. Further evidence for this is provided by Figure 4.11b which shows the error in ego-motion prediction during supervised fine-tuning where each trace represents the supervised training being started at a different stage on unsupervised training.

In Fig. 4.11b we analyse the effect of unsupervised CCA-based motion-feature learning. The test shows that the unsupervised pre-training is very effective and has a significant impact on the accuracy obtained by the final model when trained for

a set number of epochs. This result is significant as it means that the deep CCA approach is able to extract meaningful representations across sensor streams with no intervention from the engineer. This implies that during the training process it must be learning to not only filter out unuseful information in the traditional sense (i.e. noise), but also at a higher, semantic level where it builds feature representations that contain information common to both sensor streams.

A depiction of the features learned by the network (after supervised training) is shown in Fig. 4.12 which was extracted from the CNN component of the network before concatenation. It is evident that the features learned loosely correspond to the motion data. The features also reveal that there is quite a large degree of redundancy in the network's representation of motion.

A qualitative evaluation of the flow and depth predictions learned using the proposed photometric loss is illustrated in Figure 4.14. The left subfigure shows a motion where the car is moving towards the epipole with a radial flow field, while the right subfigure shows the car turning a corner where a relative constant flow-field is predicted across the image.

A quantitative evaluation of depth prediction obtained by our method are given in Table 4.2. In general our method produces sharper depth maps and slightly outperforms SfM-Net on KITTI. This can be attributed to our modelling of the relationship between flow, depth and motion as an instantanous (i.e. velocity based) relation rather than the coarse reprojection used by Sfm-Net [159].

Table 4.2: Numerical comparison of predicted flow to SfM-Net [159]

	Ours	SfM-Net [159]
Depth KITTI 2012	0.4	0.45
Depth KITTI 2015	0.38	0.41

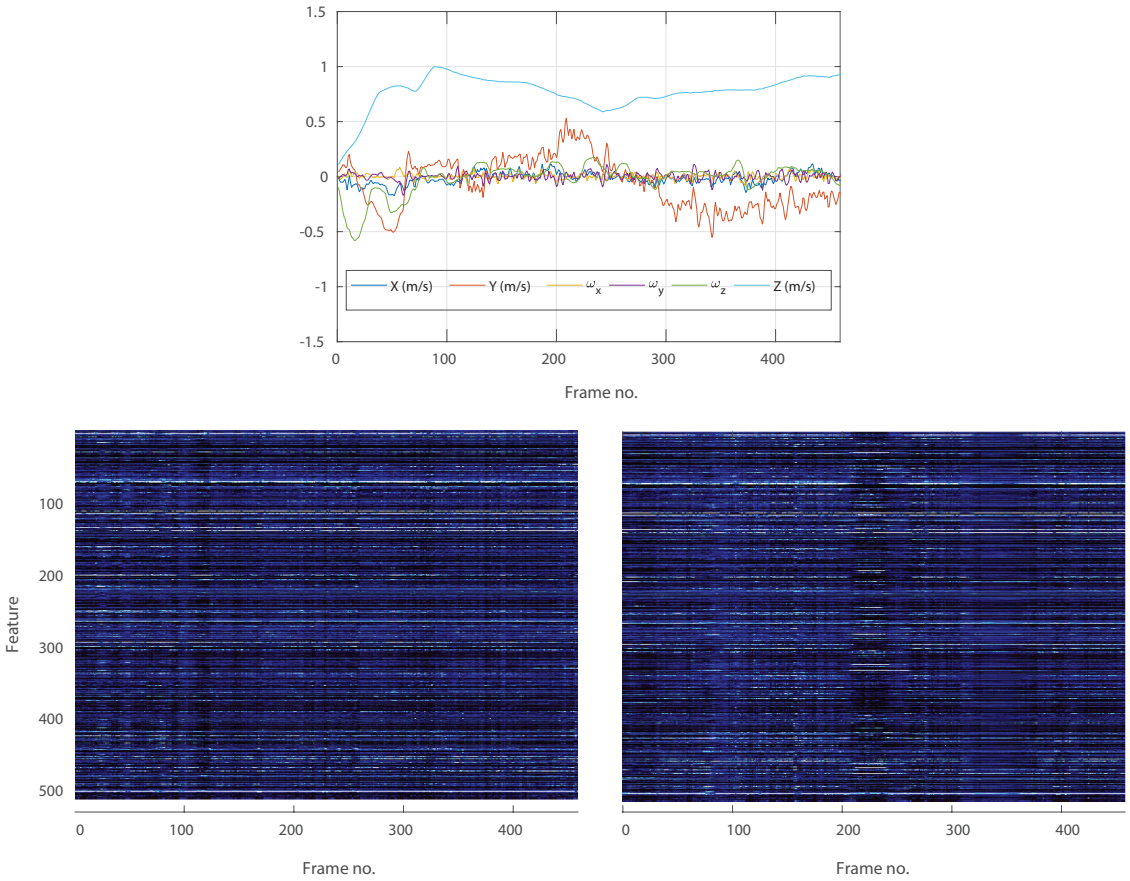


Figure 4.12: Visualization of the IMU and visual motion feature vectors over a sequence of 550 frames.

4.11 Results

4.11.1 Effectiveness of se(3) and SE(3) Joint Training

The training performance in Fig. 4.7 shows the difference between training solely on the F-2-F displacements, solely on the full $SE3$ pose and using our joint training method. The results show that joint training allows the network to converge more quickly towards low-error estimates over the training and validation sequences, while the F-2-F training converges very slowly and training on the full pose converges to a high-error estimate.

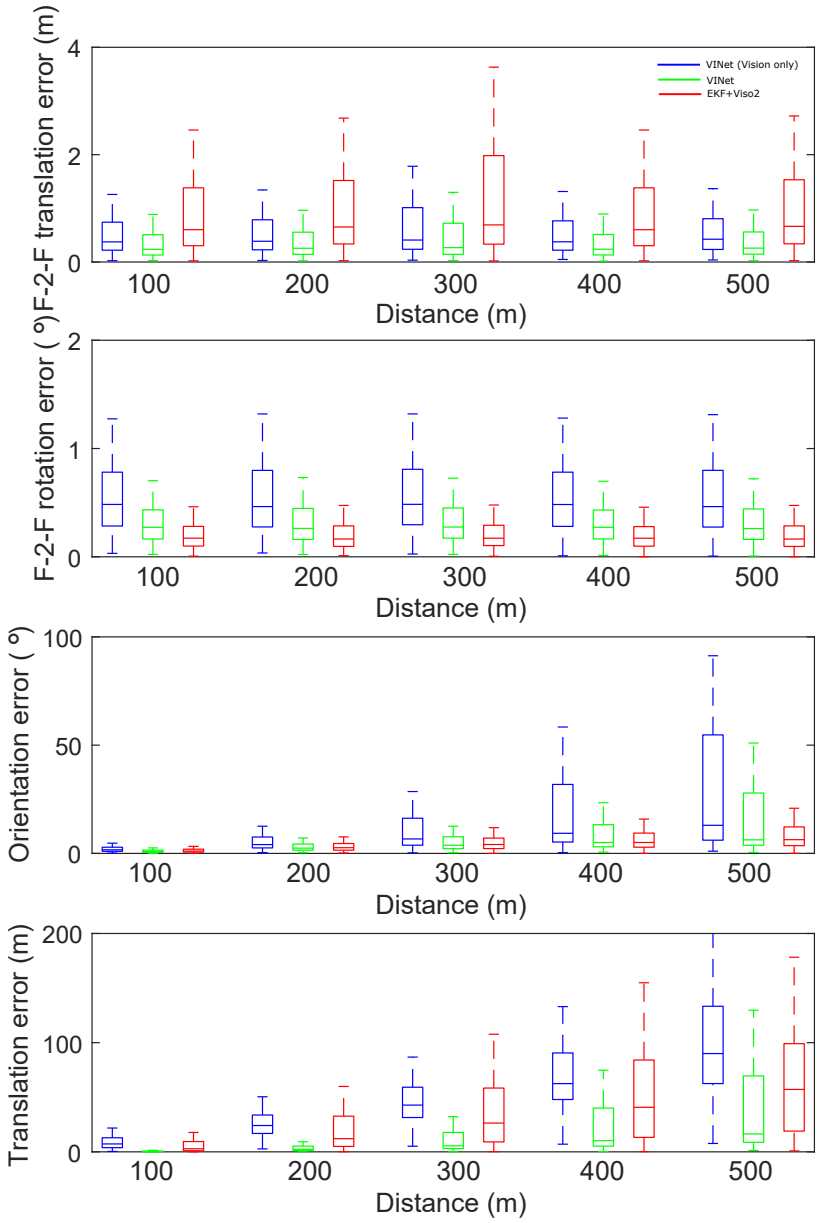


Figure 4.13: Translation and orientation errors on the KITTI dataset. Method A: Proposed (only image data), Method B: Proposed (visual-inertial data), Method C: Viso2

4.11.2 Accuracy compared to State-of-the-Art

Fig. 4.13 shows the translation and orientation errors obtained on the KITTI dataset. As expected, the visual-inertial network outperforms the network using visual data alone. The proposed also outperforms the VISO2 method in terms of translational

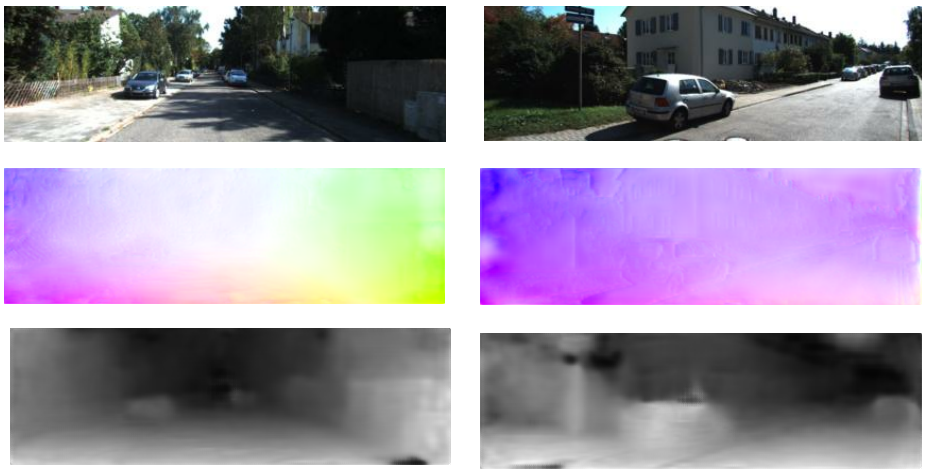


Figure 4.14: Qualitative result of the self-supervised learning on the KITTI dataset

error, however it suffers somewhat from estimating orientation where the IMU-Viso2 approach performs better. The high translational accuracy of the proposed method compared to its orientation estimation can possibly be attributed to its ability to learn to predict scale from both the image data as-well as the IMU data which is not possible in traditional approaches.

The proposed approach, although requiring no explicit manual calibration of the sensors, or tuning of special parameters, performs on-par with the Visual-Inertial EKF in all three cases. In Fig. 4.15a, although there is some error along the trajectory, the accumulated end-point error is far less than the EKF approach. In Fig. 4.15b we see again that the proposed approach is better able to estimate the scale of the trajectory and clearly outperforms the EKF-based fusion method. In Fig. 4.15c, the proposed approach exhibits slightly higher error at the turn around (lat:48.962, long:8.336), however, leading to the slightly higher accumulated end-point error.

Table 4.3: Comparison between the visual and visual-inertial aided accuracy on the EuRoC dataset

Seq	Frms	Vision-Only		Visual-Inertial	
		Rot (deg/m)	Trans (m)	Rot (deg/m)	Trans (m)
MH_01	2263	1.4277	0.1803	1.4100	0.1407
MH_02	2023	1.1010	0.1564	0.9341	0.1155
MH_03	2690	4.8551	0.1512	4.6639	0.1333
MH_04	3030	3.5797	0.1189	3.4930	0.0958
MH_05	3672	7.4204	0.2208	5.7171	0.1931

4.11.3 Robustness to Extrinsic Calibration and Synchronization

We test the robustness of our method against camera-sensor calibration errors. We introduce calibration errors by adding a rotation of a chosen magnitude and random angle $\Delta R_{SC} \sim \text{vMF}(\cdot|\mu, \kappa)$ to the camera-IMU rotation matrices R_{SC}. For our VI Net we present two sets of results - one where we have augmented the training set by artificially mis-calibrated training data and one where only the calibrated data has been used to train the network.

Fig. 4.8 shows the comparison of the estimated MAV trajectory by OKVIS and the proposed method for various levels of mis-calibration. It is evident that even when trained using no augmentation, the neural network degrades more gracefully in the face of mis-calibrated sensor data.

Table 4.4: Robustness of the VI-Network to camera-IMU calibration errors.

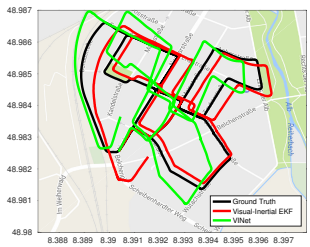
	0°	5°	10°	15°
Proposed (no-aug)	0.1751	0.8023	1.94	3.0671
Proposed (w/ aug)	0.1842	0.1951	0.2218	0.5178
OKVIS	-	-	-	-
ROVIO	0.152	0.186	0.192	0.364

Numerical results for the robustness test is shown in Tbl. 4.4. The proposed method is trained using no augmentation performs competitively compared to OKVIS

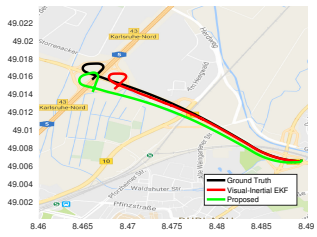
and does not fail with high calibration errors. The results for the proposed trained using calibration augmentation show a significantly hindered decrease in accuracy as the calibration errors increase. For the time-synchronization, a similar trend is observed compared to the extrinsic calibration tests. This indicates that simply by training the network using mis-calibrated data, it can be made robust to mis-calibration errors. This property is unique to the network-based approach and rather surprising as it is very difficult, if not impossible, to increase the robustness of traditional approaches in this manner.

Table 4.5: Accuracies obtained on the KITTI dataset using only visual data.

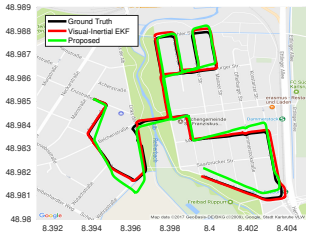
Sequence		VISO2-Stereo		[147]		Ours	
Seq	Frms	Rot	Trans	Rot	Trans	Rot	Trans
		(deg/m)	(%)	(deg/m)	(%)	(deg/m)	(deg/m)
0	4540	0.0109	2.32	0.0048	2.04	0.0037	3.2877
2	4660	0.0074	2.01	0.0035	1.50	0.0025	3.0650
3	800	0.0107	2.32	0.0021	3.37	0.0052	2.8490
4	270	0.0081	0.99	0.0023	1.43	0.0031	3.2741
5	2760	0.0098	1.78	0.0038	2.19	0.0061	2.6577
6	1100	0.0072	1.17	0.0081	2.09	0.0032	2.9796
8	4070	0.0104	2.35	0.0044	2.37	0.0077	2.9517
9	1590	0.0094	2.36	0.0047	1.76	0.0045	3.0638
10	1200	0.0086	1.37	0.0085	2.12	0.0048	2.8273
Avg		0.0094	1.852	2.06	2.096	0.0045	2.995



(a) KITTI VIO Trajectory 2011-10-30 27



(b) KITTI VIO Trajectory 2011-10-30 28



(c) KITTI VIO Trajectory 2011-10-30 42

4.12 Conclusion

In this chapter we have presented an end-to-end trainable system for ego-motion estimation using visual-inertial sensors. We have also introduced two unsupervised losses which can be used in addition to the labelled loss as a training signal for the network. The photometric loss, derived from the depth prediction and the optical flow allows for fully unsupervised learning of motion. The correlation-maximization objective, although it requires supervised fine-tuning on labelled data, is a viable feature-learning method when visual and inertial sensors are available. The goal of these methods is not to outperform the labelled training, but merely to provide a viable alternative for pre-training which we have shown is possible in this chapter.We have shown that it performs on-par with traditional approaches which require much hand-tuning during setup. Compared to traditional methods, it has the key advantage of being able to learn to become robust to calibration errors. We believe that the proposed approach is a first step towards truly robust visual-inertial sensor fusion. For future work we intend to further investigate the feature representation learned by the network as well as to incorporate more than two sensors.

Chapter 5

Learning to Re-Localize Image Streams

Machine learning techniques, namely convolutional neural networks (CNN) and regression forests, have recently shown great promise in performing 6-DoF localization of monocular images. However, in most cases image-sequences, rather only single images, are readily available. To this extent, none of the proposed learning-based approaches exploit the valuable constraint of temporal smoothness, often leading to situations where the per-frame error is larger than the camera motion. In this chapter we propose a recurrent model for performing 6-DoF localization of video-clips. We find that, even by considering only short sequences (20 frames), the pose estimates are smoothed and the localization error can be drastically reduced. Finally, we consider means of obtaining probabilistic pose estimates from our model. We evaluate our method on openly-available real-world autonomous driving and indoor localization datasets.

5.1 Related publications

The publications arising from the work in this chapter include

- **Clark, R.**, Sen, W., Wen, H., Trigoni. N. "A Deep Spatio-Temporal Model for 6-DoF Video-Clip Relocalization." IEEE International Conference on Computer Vision and Pattern Recognition, (CVPR), 2017.

5.2 Introduction

Localization of monocular images is a fundamental problem in computer vision and robotics. Camera localization forms the basis of many functions in computer vision where it is an important component of the Simultaneous Localization and Mapping (SLAM) process and has direct application, for example, in the navigation of autonomous robots and drones in first-response scenarios or the localization of wearable devices in assistive living applications.

The most common means of performing 6-DOF pose estimation using visual data is to make use of specially-built models, which are constructed from a vast number of local features that have been extracted from the images captured during mapping.

The 3D locations of these features are then found using a Structure-from-Motion (SfM) process, creating a many-to-one mapping from feature descriptors to 3D points. Traditionally, localizing a new query image against these models involves finding a large set of putative correspondences. The pose is then found using RANSAC to reject outlier correspondences and optimize the camera pose on inliers. Although this traditional approach has proven to be incredibly accurate in many situations, it faces numerous and significant challenges. These methods rely on local and unintuitive hand-crafted features, such as SIFT keypoints. Because of their local nature, establishing a sufficient number of reliable correspondences between the image pixels and the map is very challenging. Spurious correspondences arise due to both "well-behaved" phenomena such as sensor noise and quantization effects as well as pure outliers which arise due to the local correspondence assumptions not being satisfied

[66]. These include inevitable environmental appearance changes due to, for example, changing light levels or dynamic elements such as clutter or people in the frame or the opening and closing of doors. These aspects conspire to give rise to a vast number of suprious correspondences, making it difficult to use for any purpose but the localization of crisp and high-resolution images. Secondly, the maps often consists of millions of elements which need to be searched, making it very computationally intensive and difficult to establish correspondences in real-time.

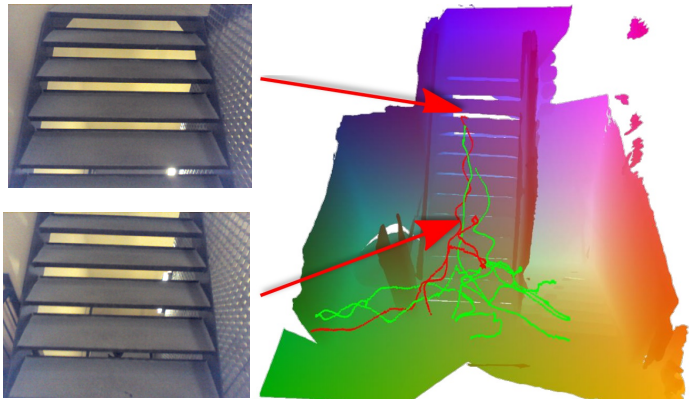


Figure 5.1: An extreme example of perceptual aliasing in the Stairs scene of the Microsoft 7-Scenes dataset. One of the frames is taken at the bottom of the staircase and the other near the top. Using only single frames, as in the competing approaches, it would be impossible to correctly localize these images.

Recently, however, it has been shown that machine learning methods such as random forests [142] and convolutional neural networks (CNNs) [83] have the ability to act as a regression model which directly estimates pose from an input image with no expensive feature extraction or feature matching processes required. These methods consider the input images as being entirely uncorrelated and produce independent pose estimates that are incredibly noisy when applied to image sequences. On most platforms, including smart-phones, mobile robots and drones, image-sequences are readily obtained and have the potential to greatly enhance the accuracy of these approaches and promising results have been obtained for sequence-based learning for relative pose estimation [24]. Therefore, in this chapter we consider ways in which we

can leverage the temporal dependencies in image-sequences to improve the accuracy of 6-DoF camera re-localization. Furthermore, we show how we can in essence unify map-matching, model-based localization, and temporal filtering all in one, extremely compact model.

5.2.1 Related Work

Map-matching Map matching methods make use of a map of a space either in the form of roads and traversable paths or a floor-plan of navigable and non-navigable areas to localize a robot as it traverses the environment. Map-matching techniques are typified by their non-reliance on strict data-association and can use both exteroceptive (eg. laser scans) or interoceptive (odometry, the trajectory or the motion of the platform) sensors to obtain a global pose estimate. The global pose estimate is obtained through probabilistic methods such as sequential Monte Carlo (sMC) filtering [68] or hidden Markov models (HMMs) [117]. These methods inherently incorporate sequential observations, but accuracy is inferior to localizing against specialised maps, such as a 3D map of sparse features.

Sparse feature based localization When a 3D model of discriminative feature points is available (eg. obtained using SfM) then the poses of query images can be found using camera re-sectioning. Matching against large 3D models is generally very computationally expensive and requires lots of memory space to store the map. A number of approaches have been proposed to improve the efficiency of standard 3D-to-2D feature matching between the image and the 3D model [167]. For example, [135] propose a quantized feature vocabulary for direct 2D-to-3D matching with the camera pose being found using RANSAC in combination with a PnP algorithm and in [136] an active search method is proposed to efficiently find more reliable correspondences. [103] propose a client-server architecture where the client exploits sequential images to perform high-rate local 6-DoF tracking which is then combined with lower-rate

global localization updates from the server, entirely eliminating the need for loop-closure. The authors propose various methods to integrate the smooth local poses with the global updates. In [87] the authors consider means of improving the global accuracy by introducing temporal constraints into the image registration process by regularizing the poses trough smoothing.

Scene coordinate regression forests of Shotton et al. [142] use a regression forest to learn the mapping between the pixels of an RGB-D input image and the scene co-ordinates of a previously established model. In essence the regression forest learns the function $f : (r, g, b, d, u, v) \rightarrow (U, V, W)$. To perform localization, a number of RGB-D pixels from the query image are fed through the forest and a RANSAC-based pose computation is used to determine a consistent and accurate final camera pose. To account for the temporal regularity of image sequences, the authors consider a frame-to-frame extension of their method. To accomplish this, they initialize one of the pose hypotheses with that obtained from the previous frame, which results in a significant improvement in localization accuracy. Although extremely accurate, the main disadvantage of this approach is that it requires depth images to function and does not eliminate the expensive RANSAC procedure.

CNN features Deep learning is quickly becoming the dominant approach in computer vision. The many layers of a pre-trained CNN form a hierarchical model with increasingly higher level representations of the input data as one moves up the layers. It has been shown that many computer vision related tasks benefit from using the output from these upper layers as feature representations of the input images. These features have the advantage of being low-level enough to provide representations for a large number of concepts, yet are abstract enough to allow these concepts to be recognized using simple linear classifiers [140]. They have shown great success applied to a wide range of tasks including logo classification [8], and more closes related to our goals, scene recognition [169] and place recognition [152].

Posenet [83] demonstrated the feasibility of estimating the pose of a single RGB image by using a deep CNN model to regress directly on the pose. For practical camera relocalization, Posenet is far from ideal. For example, on the Microsoft 7-Scenes dataset it achieves a $0.48m$ error where the model space is only $2.5m \times 1m \times 1m$. A limitation of the PoseNet architecture is that it can only produce a single pose estimate at a time. Our approach is a multi-frame extension of PoseNet which enhances the localization accuracy by incorporating a temporal aspect in the model, producing smoother and more accurate estimates.

5.2.2 Contributions

In this chapter, we propose a recurrent model for reducing the pose estimation error by using multiple frames for the pose prediction. Our specific contributions are as follows:

1. We present a deep spatio-temporal model for efficent global localization from a monocular image sequence.

2. We integrate into our network a method for obtaining the instantaneous covariances of pose estimates.

3. We evaluate our approach to two large open datasets and show that the proposed spatio-temporal model significantly outperforms a smoothing baseline.

5.3 Proposed Model

In this section we outline our proposed model for video-clip localization, VidLoc, a high-level overview of which is shown in Figure 5.2. Our model processes the video image frames using CNN and integrates temporal information through a bidirectional LSTM.

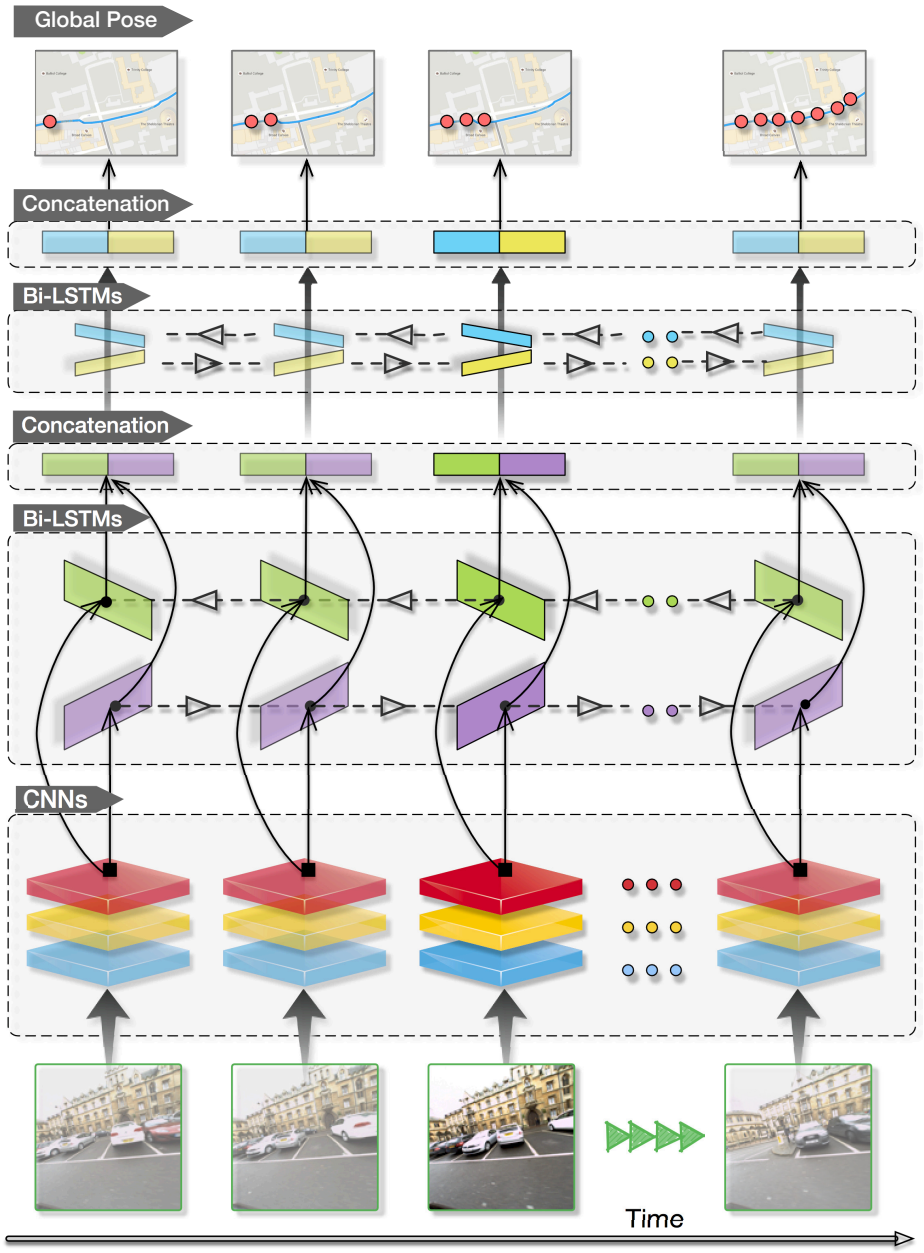


Figure 5.2: The CNN-RNN network for video-clip localization. The Bidirectional LSTM contains multiple layers (in this case two) which operate similar to a coarse-to-fine structure, where the first layers predict the large scale structure of the trajectory and the later layers refine these predictions.

5.3.1 Image Features: CNN

The goal of the CNN part of our model is to extract relevant features from the input images that can be used to predict the global pose of an image. A CNN consists

of stacked layers performing convolution and pooling operations on the input image. There are a large number of CNN architectures that have been proposed, most for classifying objects in images and trained on the Imagenet database. These models, however, generalize well to other tasks, including pose estimation. As in the Posenet [83] paper, VGGNet [145] is able to produce more accurate pose estimates, but incurs a high-computational cost due to its very deep architecture. As we are interested in processing multiple images in a temporal sequence we adopt the GoogleNet Inception [154] architecture for the VidLoc CNN. We use only the convolutional and pooling layers of GoogleNet and drop all the fully-connected layers. In our experiments, we explore the impact on computational efficiency incurred vs. the increase in accuracy obtained using multiple frames.

5.3.2 Temporal Modelling: Bidirectional RNN

In Posenet and many other traditional image based localization approaches, the pose estimates are produced entirely independently for each frame. However, when using image-streams with temporal continuity, a great deal of pose information can be gained by exploiting the temporal dependencies. For example, adjacent images often contain views of the same object which can boost the confidence in a particular location, and there are also tight constraints on the motion that can be undergone in-between frames - a set of frames estimated to be at a particular location are very unlikely to contain one or two located far away.

To capture these dynamic dependencies, we make use of the LSTM model in our network. The LSTM [71] extends standard RNNs to enable them to learn long-term time dependencies. This is accomplished by including a *forget gate*, input and output *reset gates* and a *memory cell*. The flow of information into and out-of the memory cell is regulated by the forget and input gates. This allows the network to overcome the vanishing gradient problem during training and thereby allow it to learn long-

term dependencies. The input to the LSTM is the output of the CNN consisting of a sequence of feature vectors, $\mathbf{x}_t$. The LSTM maps the input sequence to the output sequence consisting of the global pose parameterised as a 7-dimensional vector, $\mathbf{y}_t$ consisting of a translation vector and orientation quaternion. The activations of the LSTM are computed by interatively applying the following operations on each timestep

$$\begin{aligned} f_t &= \sigma_g(W_f\mathbf{x}_t + U_f\mathbf{h}_{t-1} + b_f) \\ i_t &= \sigma_g(W_i\mathbf{x}_t + U_i\mathbf{h}_{t-1} + b_i) \\ o_t &= \sigma_g(W_o\mathbf{x}_t + U_o\mathbf{h}_{t-1} + b_o) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c\mathbf{x}_t + U_c\mathbf{h}_{t-1} + b_c) \\ \mathbf{h}_t &= o_t \circ \sigma_h(c_t) \\ \mathbf{y}_t &= \sigma_o(W_y h_t + b_y) \end{aligned} \tag{5.1}$$

where W, U and b are the parameters of the LSTM, f_t, i_t, o_t are the gate vectors, σ_g is the non-linear activation function and h_t is the hidden activation of the LSTM. For the inner activations, we use a hyperbolic tangent function and for the output σ_o we use a linear activation. A limitation of the standard LSTM model is that it is only able to make use of previous context in predicting the current output. For our monocular image-sequence pose prediction application we have a sliding window of frames available at any one instance in time and thus we can exploit both future and past contextual information predicting the poses for each frame in the sequence. For this reason, we adopt a Bidirectional architecture [139] for our LSTM model. The bidirectional model assumes the same state equations as in 5.1, but uses both future and past information for each frame by using two hidden states, $\overleftarrow{\mathbf{h}}_t$ and $\overrightarrow{\mathbf{h}}_t$, one for processing the data forwards and and the other for processing backwards, as shown in Figure 5.3. The hidden states are then combined to form a single hidden state $\mathbf{h}_t$

through a concatenation operation

$$\mathbf{h}_t = \left[\overleftarrow{\mathbf{h}_t}, \overrightarrow{\mathbf{h}_t}\right] \tag{5.2}$$

The output pose is computed from this hidden layer as in 5.1.

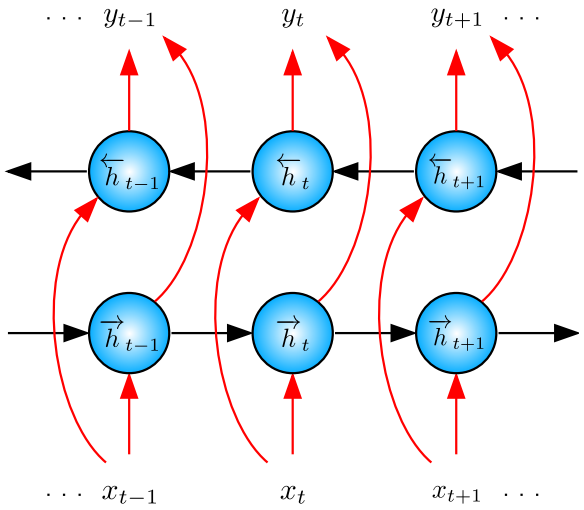


Figure 5.3: The structure of a bidirectional RNN [139].

5.3.3 Network Loss

In order to train the network we use the sum of the Euclidean error magnitude of both the translation and orientation. To compute the loss, we separate the output of the LSTM into the translation $\mathbf{x}_t$ and orientation $\mathbf{q}_t$

$$\mathbf{y}_t = [\mathbf{x}_t, \mathbf{q}_t] \tag{5.3}$$

and use a weighted sum of the error magnitudes of the two component vectors

$$\mathcal{L} = \sum_{t=1}^{T} \alpha_1 ||\mathbf{x}_t - \hat{\mathbf{x}}_t|| + \alpha_2 ||\mathbf{q}_t - \hat{\mathbf{q}}_t|| \tag{5.4}$$

We propagate the loss through the temporal frames in each training sequence by

unrolling the network and performing back-propagation through time. To update the weights of the layers, we make use of the Adam optimizer.

5.3.4 Probabilistic Pose Estimates

Pose estimation methods, no matter how accurate, will always be subject to a degree of uncertainty. Being able to correctly model and predict uncertainty is thus a key component of any useful visual localization method. The euclidean sum-of-squares error which we defined in Sec. 5.3.3 results in a network which approximates only the uni-modal conditional mean of the pose as defined by the training data. In essence the output of the network can be regarded as predicting $\mu_{\mathbf{x}}$, the mean of the conditional pose distribution $p\left([\mathbf{x}, \mathbf{q}]|I\right) = \mathcal{N}\left(\mu_{[\mathbf{x},\mathbf{q}]}, \sigma\right)$ where the Guassian assumption is induced by the use of the square error loss. For the unlikely case where the actual posterior pose distribution is Gaussian, this mean represents the optimal distribution in a maximum-likelihood sense. However, for global camera re-localization as we are concerned with in this thesis, this assumption is unlikely. In many instances the appearance of a space is similar at multiple locations, for example, two corridors in a building may appear very similar (known as the "perceptual aliasing" problem and in most instances cannot be addressed using visual data alone).

In [82], one possible means of representing multi-modal uncertainty in the global pose estimation was considered. In this work, the authors create a Bayesian convolutional neural network by using dropout as a means of sampling the model weights. The posterior distribution of the model weights $p\left(\mathbf{W}|\mathbf{X}, \mathbf{Y}\right)$ is intractable and they use variational inference to approximate it as proposed in [58]. To produce probabilistic pose estimates, Monte Carlo pose samples are drawn and the mean and variance determined from these. Although this models the uncertainty in the *model weights* correctly (i.e. the distribution of the model weights according to the training data), it does not fully capture the uncertainty of the pose estimates.

To model the pose uncertainty, we adopt the mixture density networks method [10]. This approach replaces the Gaussian with a mixture model, allowing a multi-modal posterior output distribution to be modelled. Using this approach, the pose estimates now take the form

$$p\left([\mathbf{x},\mathbf{q}]|I\right) = \sum_{i=1}^{M} \alpha_i(I)\mathcal{N}_i\left(\mu_{[\mathbf{x},\mathbf{q}]}(I), \sigma(I)\right) \tag{5.5}$$

where $\mathcal{N}_i\left(\mu_{[\mathbf{x},\mathbf{q}]}, \sigma|I\right)$ is a mixture component and α_i are the coefficients of the mixture distribution which satisfy the constraint $\sum_i \alpha_i = 1$. The mixing components are a function of the input image which is modelled by the network. As in the single Gaussian case, the network is trained to maximize the likelihood of the training data.

5.4 Experiments

In this section, the proposed approach is evaluated on outdoor and indoor datasets by comparing with the state-of-the-art methods.

5.4.1 Datasets

Two well-known public datasets are employed in our experiments. They demonstrate indoor human motion and outdoor autonomous car driving scenarios, respectively.

The first is the Microsoft 7-Scenes Dataset which contains RGB-D image sequences of 7 different indoor environments [142], created by using a Kinect sensor. It has been widely used for camera tracking and relocalization [83]. The images were captured at 640×480 resolution with ground truth from KinectFusion system. Since there are several image sequences of one scene and each sequence is composed of about 500-1000 image frames, it is ideal for our experiments. Ground truth camera poses for the dataset are obtained using the KinectFusion algorithm [115] to produce smooth camera tracks and a dense 3D model of each scene. In our experiments, all the 7

scenes are adopted to evaluate the proposed method. We use the same Train and Test split of the sequences as used in the original paper. This dataset consists of both RGB and depth images. Although we focus mainly on RGB-only localization, our method extends naturally to the RGB-D case.

In order to further test the performance in large-scale outdoor environments, the recently released Oxford RobotCar dataset [100] is used. It was recorded by using an autonomous Nissan LEAF car traversing in the central Oxford for a year period. The dataset contains high-resolution images from a Bumblebee stereo camera, LiDAR scanning, and GPS/INS. Since different weather conditions, such as sunny and snowy days, are exhibited in the dataset, it is very challenging for some tasks based on vision, e.g., global localization and loop closure detection across long terms and seasons. Because global re-localization does not need to have high-frequency images, the frame rate is about 1Hz in our robotcar experiments.

5.4.2 Competing algorithms

In this section we describe the experiments that we performed on the Microsoft 7-Scenes dataset. We compare our approach to the current state-of-the-art monocular camera localization methods.

Smoothing baseline The traditional means of integrating temporal information is to perform a filtering or smoothing operation on the independent pose predictions for each frame. We thus compare our method to a smoothing operation in order to investigate the advantage of using an RNN to capture the temporal information and whether global pose accuracies obtained for each frame are indeed more accurate than independent pose predictions. For our smoothing baseline, we use the spline fitting approach as per [87].

Posenet Posenet uses a CNN to predict the pose of an input RGB image. The Posenet network is the GoogleNet architecture with the top-most fully connected

layer removed and replaced by one with a 7-dimensional output and trained to predict the pose of the image.

Score-Forest The Score-Forest [142] approach trains a random regression forest to predict the scene coordinates of pixels in the images. A set of predicted scene coordinates is then used to determine the camera pose using a RANSAC-loop. We use the open source implementation for our experiments[1].

Additional comparisons Comparison to [135]. For RobotCar we extract SURF features and assign 3D locations using LiDAR data. We also provide a comparison on 7 Scenes to [13] from the results as presented in [13].

5.4.3 Experiments on Microsoft 7-Scenes Dataset

Table 5.1: Comparison to state-of-the-art approaches to monocular camera localization

Scene	Spatial Extent (m)	Score Forest	Posenet	Bayesian Posenet	BL	VidLoc	VidLoc RGB-D	VidLoc Depth
Chess	3x2x1	0.03m	0.32m	0.37m	0.32m	0.18m	0.16m	0.19m
Office	2.5x2x1.5	0.04m	0.48m	0.48m	0.38m	0.26m	0.24m	0.32m
Fire	2.5x1x1	0.05m	0.47m	0.43m	0.45m	0.21m	0.19m	0.22m
Pumpkin	2.5x2x1	0.04m	0.47m	0.61m	0.42m	0.36m	0.33m	0.15m
Kitchen	4x3x1.5	0.04m	0.59m	0.58m	0.57m	0.31m	0.28m	0.38m
Stairs	2.5x2x1.5	0.32m	0.47m	0.48m	0.44m	0.26m	0.24m	0.27m
Heads	2x0.5x1	0.06m	0.29m	0.31m	0.19m	0.14m	0.13m	0.27m
Average		0.082m	0.44m	0.465m	0.39m	0.22m	0.31m	0.26m

The results of our experiments testing the accuracy of our method are shown in Table 5.1. The proposed method significantly outperforms the Posenet approach in all of the test scenes, resulting in a $23.4\% - 55\%$ increase in accuracy. The SCoRe-forest outperforms the RGB-only VidLoc. However, this is strictly not a fair comparison for two reasons: firstly, SCoRe-forest requires depth images as input; secondly, the SCoRe-forest sometimes produces pose estimates with gross errors although these are

[1] https://github.com/ISUE/relocforests

Table 5.2: Additional comparisons on RobotCar and 7 Scenes. (1) Sparse RGB, (2) PoseNet, (3) Proposed, (4) Brachmann et al., (5) Sattler et al. [15] .

	$5cm, 5°$	Avg. Error
1	40.7%	-
2	-	46.9cm, 5.4°
3	-	25.7cm, 3.8°
4	55.2%	6.1cm, 2.7°
5	N/A	6cm, 2.89°

rejected by the RANSAC-loop, which means that pose estimates are not available for all frames. In contrast, our method produces reliable estimates for the entire sequence.

We tested our method using both depth and RGB input and although our method seamlessly utilises the depth images when available, a disadvantage is that it cannot utilise the depth information to the extent that the SCoRe-Forest is able to. This is evidenced in the accuracy results reported in Table 5.1 where it can be seen that although our method consistently achieves centimeter accuracy, it does not outperform the SCoRe-Forest. This is surprising but perhaps indicative of the operation of the network. This suggests that the network learns to perform pose prediction in a similar fashion to an appearance based localization method. In this manner, it uses both the RGB and the depth information in the same way. This is in contrast to the SCoRe-forest approach where the depth information is explicitly used in a geometric pose computation by means of the PnP algorithm. We note, however, that our method still has the advantage of being able to operate on RGB data when no depth information is available and is able to produce global pose estimates for all frames whereas the SCoRe-forest cannot.

Effect of sequence length A key result of this chapter is shown in Figure 5.5 which depicts the localization error as a function of the sequence length used. We have trained the models using sequence lengths of 200 frames in order to test the ability

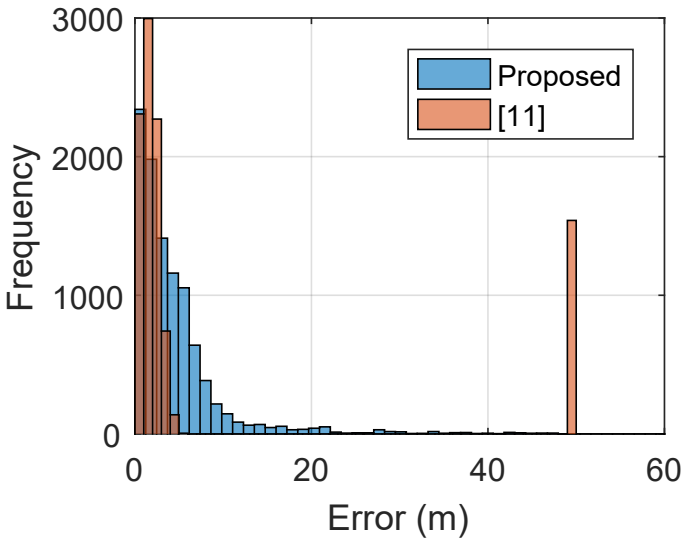


Figure 5.4: Error histogram of VidLoc compared to a sparse-feature based method [135] on the RobotCar dataset.

of the model to generalize to longer sequences. In all cases we ensure that the error is averaged over the same number and an even distribution across the test sequence. As expected, increasing the number of frames improves the localization accuracy. We also see that the model is able to generalize to longer sequences (i.e we still get an improvement in accuracy for sequence lengths greater than 200). At very long sequence lengths we experience diminishing returns - however this is not necessarily a product of the models inability to use this data but rather the actual utility of very long-term dependencies in predicting the current pose.

Timings Our approach improves on the accuracy of Posenet, yet has very little impact on the computational time. This is because processing each frame only relies on the hidden state of the RNN from the previous time instance and image data of the current frame. Predicting a pose thus only requires a forward pass of the image through the CNN and propagating the hidden state. On our test machine with a Titan X Pascal GPU, this takes only 18ms using GoogleNet and 43ms using a VGG16 CNN. An interesting observation from our experiments is that the training time to

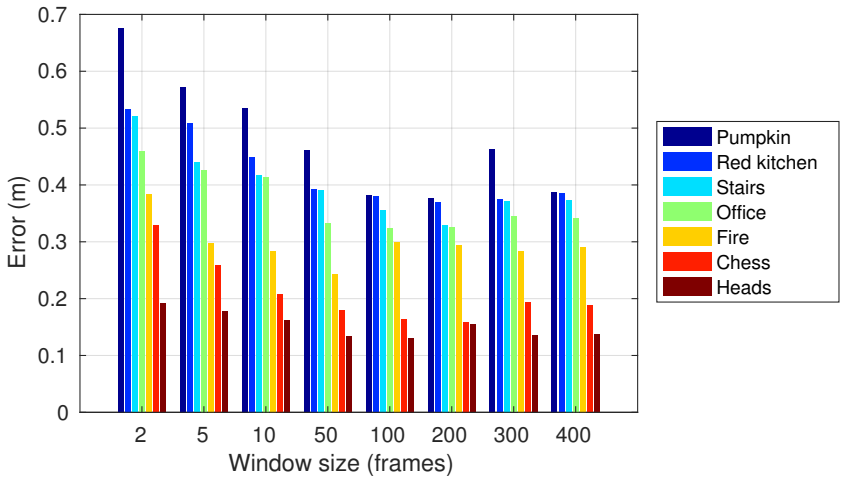


Figure 5.5: The effect of window length on pose accuracy for the sequences in the Microsoft 7-Scenes dataset.

create a usable localization network using the fine-tuning approach with Imagenet initialization is actually rather short. Typically convergence time (to around 90%) of final accuracy on the test data is around 50s.

Uncertainty output The 7-Scenes indoor dataset is extremely challenging, mainly due to the problem of perceptual aliasing as shown in Figure 5.1. One image was taken from the bottom of the staircase while the other was taken near the top. For comparing the uncertainty to [82] we use the Bayesian PoseNet implementation provided by the authors of [82]. For experimentation we use a single Gaussian component in the mixture model. To achive better convergence in the training procedure we first train the mean, while fixing the variance and then relax the variance afterwards. Without following this training method we found the the Gaussian tends to drift and not converge very quickly. Figure 5.6 shows a visualization of the predicted uncertainty and the actual error in the format of [82]. The percentage of pose errors fall within the 3σ bound for the proposed adopted uncertainty method is 97.2% and [82] is 98.1% (this value is ideally 99.7%). Both approaches produce high-quality uncertainty estimates, although the proposed is a bit less conservative. The proposed requires no approxi-

mation or sampling. However, in many cases we found that the predicted variance is rather high and we leave it as future work to improve the variance prediction.

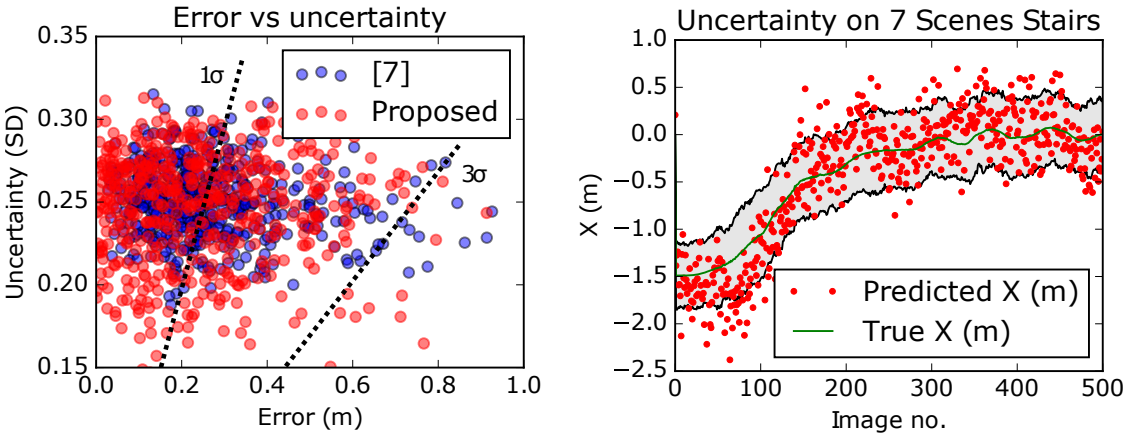


Figure 5.6: (a)Comparison of uncertainty to [7] and (b) visualization of proposed uncertainty prediction (1σ) and trajectory.

5.4.4 Experiments on RobotCar Dataset

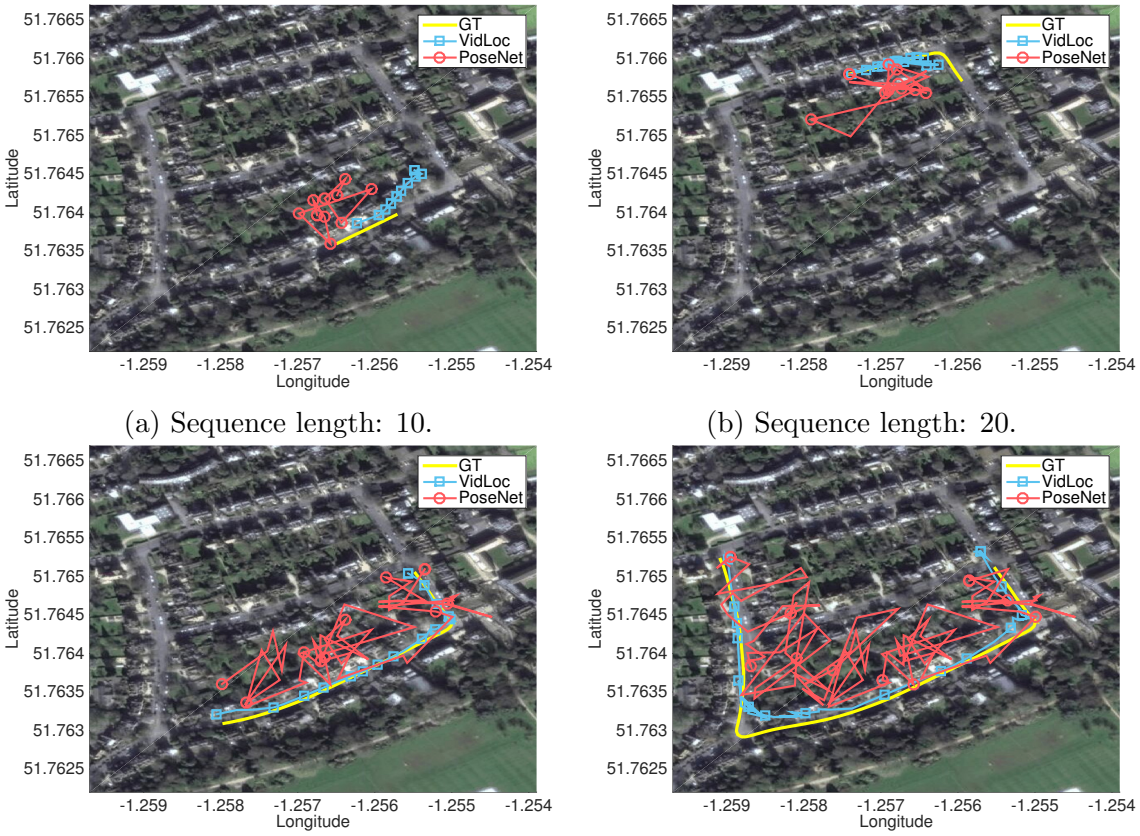
The experiments on the Oxford RobotCar Dataset are given in this section. Since the GPS/INS poses are relatively noisy (zig-zag track), they are fused with stereo visual odometry by using pose graph SLAM to produce smooth ground truth for training. In our experiments, three image sequences are used for training, while the trained models are tested on another new testing sequence.

(a) Same location at different times. (b) Different locations but close times.

Figure 5.7: Localizing in real-world scenes is incredibly challenging partly due to the ambiguity of the appearance of locations.

The image sequences selected are very challenging for global re-localization. As shown in Figure 5.7a, the images are mostly filled with roads and trees, which do not

have distinct and consistent appearance features. Specifically, three images of a same location yet captured at different times are presented in Figure 5.7a. Although they are taken at a same position, the cars parking along the road introduce significant appearance changes. Without viewing the buildings around, the only consistent objects which can be useful for global re-localization are the trees and roads. However, they are subtle in terms of image context. For example, Figure 5.7b shows sample images of three different locations which share very similar appearance. Again, this perceptual aliasing makes global re-localization more challenging by only using one single image.



(a) Sequence length: 10.

(b) Sequence length: 20.

(c) Sequence length: 50.

(d) Sequence length: 100.

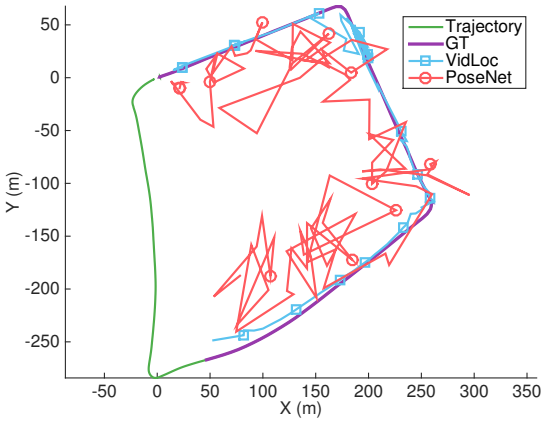
Figure 5.8: Visual depiction of the errors on the robotcar dataset.

The global re-localization results of the testing image sequence with lengths 10, 20, 50 and 100 are shown in Figure 5.8a,5.8b,5.8c against ground truth. They are

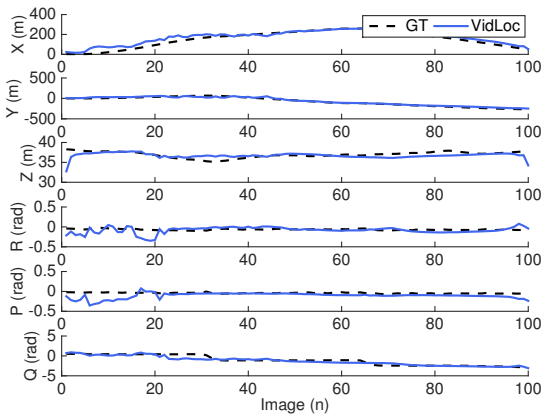
also superimposed on Google Map. It can be seen that the result of the proposed method improves as the length of the sequence increases, and the re-localization results of the lengths 50 and 100 match with the roads consistently. It is interesting to see that its trajectories are also able to track the shape of motion by end-to-end learning. In contrast, the Posenet which uses a single image suffers from noisy pose estimates around the ground truth. This experiment validates the effectiveness and necessity of using sequential images for global re-localization, mitigating the problems of perceptual aliasing and improving localization accuracy.

Localization trajectories and 6-DoF pose estimation of a sequence with 100 length are given in Figure 5.9a. It further shows that the localization result is smooth and accurate. The corresponding estimation of the 6-DoF poses on x, y, z, roll, pitch and yaw is described in Figure 5.9b. It can be seen that the proposed method can track the ground truth accurately in terms of 6-DoF pose estimation. This is of importance when using the localization result for re-localization and loop closure detection.

Figure 5.10b illustrates the distribution and histogram of the re-localization errors (mean squared errors) of all sequences with 100 length. Statistically more than half of poses estimated by the proposed method are within 20 meters, while this is less than

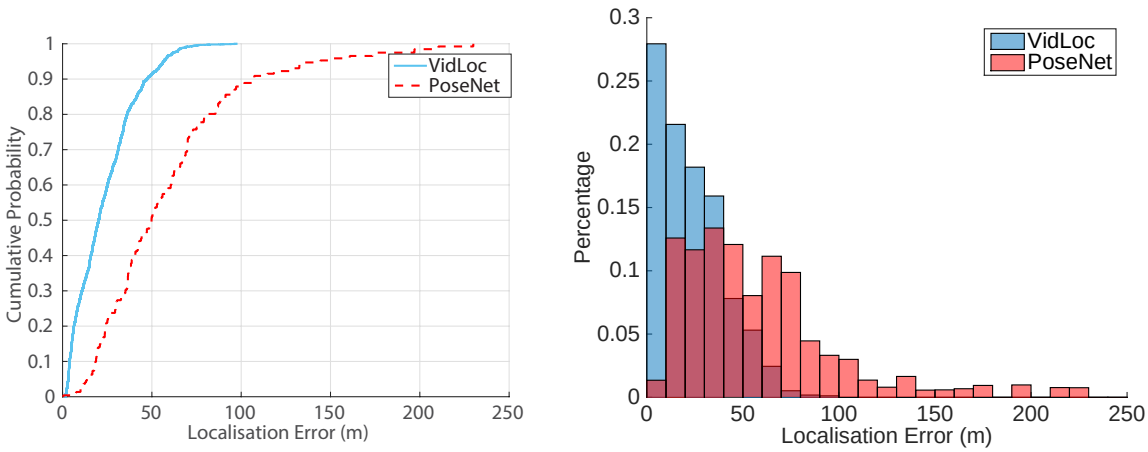


(a) Visualization of the predicted and ground-truth trajectory.



(b) Time-series plot of the poses.

Figure 5.9: Estimates of 6-DoF poses.



(a) Distribution of errors.

(b) Histogram of errors.

Figure 5.10: Histogram and CDF of the errors of VidLoc compared to PoseNet.

15% percentage for Posenet. Moreover, there are some big errors, e.g., more than 200 meters, of Posenet, which indicates that it may have perceptual aliasing problems during pose estimation. It tends to be common in this challenging dataset, as shown in Figure 5.7b. Therefore, it is verified that the recurrent model encapsulating the relationship between consecutive image frames is effective for global re-localisation using a video clip.

5.5 Conclusion

We have presented an approach for 6-DoF video-clip re-localization that exploits the temporal dependencies in the video stream to improve the localization accuracy of the global pose estimates. We have studied the impact of window size and shown that our method outperforms the closest related approaches for monocular RGB localization by a fair margin.

For future work we intend to investigate means of making better use of the depth information, perhaps by forcing the network to learn to make use of geometrical information. One means of doing this would be to try predict the scene coordinates

of the input RGB-D image using the CNN in an intermediate layer and then derive the pose from this and the input image. In essence this would be like unifying appearance-based localization and geometry-based localization in one model. We also intend to investigate better training methods for the mixture components as training with more than one component is exceedingly difficult.

Chapter 6

SLAM system integration

In the preceding chapters we introduced deep recurrent models to perform robust ego-motion and global re-localization using raw image sequences as input and described how covariance estimates can be obtained through a maximum-likelihood training approach.

The relative and global pose estimates obtained from our learning-based approaches fall short of being useful in a stand-alone system for real-world deployment. This is for 2 reasons 1) the odometry drifts over time leading to the accumulated error dominating the pose estimate and 2) the global re-localization model cannot automatically generalize to new environments and the pose predictions obtained through this model are far more inaccurate than the relative poses obtained via the odometry. Furthermore, the predictions obtained using the deep networks do not take into account the information that is specific to the images being received - commonly referred to as the "data term" in energy-based optimization approaches.

We now turn our attention to traditional visual-inertial SLAM approaches to see if we can utilize this knowledge to boost the performance of our learned system. In this regard, motion priors are an important component of traditional SLAM systems - the motion priors can be used to guide data-association, predict feature locations and

to provide an initialization for the optimization of the camera pose [165]. Even so, most traditional SLAM systems employ a simple, constant motion model to estimate the prior for the current frame [33]. For this reason we propose to use our networks as a more informative prior for the SLAM process.

The characteristics of the learned models are the perfect candidate to act as a general and robust prior for pose estimation and this is how we utilize them in our system. In this chapter we present a means of integrating these priors in an optimization framework that is able to drive drift and global error down on a per sequence basis.

In this chapter, we present an approach to integrating the learned estimates as part of a larger SLAM system in order to minimize the drift. Our approach finds a maximum-a-posteriori estimate to the robot pose and scene structure using the deep recurrent models as priors and live input from a visual-inertial sensor. Our specific technical contributions in this chapter include the novel use of deep priors which we tightly integrate with a traditional keyframe-based visual-inertial SLAM system as well as an efficient keyframe selection algorithm.

6.1 Related publications

The contributions described in this chapter are related to the following publications

- **Clark, R.** "Deep Priors for Dense Visual-Inertial SLAM" IEEE Robotics and Automation Letters (RAL), 2017. *(TBD)*
- **Clark, R.**, Wang, S., Wen, H., Trigoni. N., Markham, A. "iSfM: Pushing the Limits of 3D Indoor Modelling using Mobile Sensing." IEEE Transactions on Mobile Computing (TMC).

6.2 Proposed system

The proposed method constitutes a keyframe-based SLAM system. To acquire images, the user holds the camera in front of them and walks across the space to be modelled. Keyframes are selected efficiently on the client side, by intelligently sampling images that will best aid the reconstructions. This is accomplished through a adaptive frame selection component (Sec. 6.5), which is implemented in realtime on the device and serves the purpose of selecting keyframes. The selected images, along with the IMU data, are then sent to the optimization for computing MaP estimates of the pose and feature locations. The proposed SLAM is robust to reconstruction errors by tightly integrating the prior estimates on the relative motion as well as the feature locations by utilising the output of a single-image depth prediction neural network. The optimization performs a MaP model-inference which ensures reconstructions that are consistent with all available data. As in most keyframe-based SLAM systems, the optimization is carried out over a sliding window of individual frames.

Even with the introduction of the local optimization of windows of frames, the pose estimates are still subject to drift and thus we use loop-closure detections and a SLAM graph optimization at the level of poses. The loop-closures are detected using a standard bag-of-words based approach using the DBoW2 library.

Our re-localization comprises the global pose estimates from our trained RNN model (described further in Chapter 7). We use this global pose estimate as a prior in the global SLAM graph optimization.

The entire process is detailed in Sec. 6.3. In this way, by utilising learned priors, we achieve a more accurate V-I SLAM than existing approaches, making it more attractive for widespread adoption.

6.3 Visual-Inertial Structure from Motion

The SfM process operates as follows. A window of input images are supplied to the SfM software pipeline. We run a feature detection algorithm on the images which extracts a set of distinct key-points in each image. A descriptor is extracted at each key-point, allowing key-points to be matched across images. After feature matching, we perform an optimization over the window of frames considering both the visual and inertial data. This process is less computationally expensive as it is linear in the number of feature matches across images (called tracks). This stage uses the tracks to recover the translation and orientation of each image as well as a sparse point cloud of the environment.

Given n projected points $u_{i,j}, i \in 1 \dots m, j \in 1 \dots n$ in m images, we optimize on the state of each camera matrices $x_1 \dots x_m$, that describes the camera calibration and as well as a consistent structure consisting of 3D points $X_1 \dots X_n$ representing the model reconstructed from the m images. Each image point is associated with a feature $f_{i,j}$ that is computed using BRISK features. The projection matrices consist of the camera calibration K and the pose i.e. rotation R and translation T of the image: $P = K[R|T]$.

In a traditional state-of-the-art optimization-based visual inertial odometry system the state of the system consists of a 15 dimensional vector

$$x = \begin{bmatrix} \left[\log_{SO(3)}(R_j^w)^\vee\right]^\top & p_j^{w\top} & v_j^{w\top} & b_\omega^\top & b_a^\top \end{bmatrix}^\top \tag{11}$$

The total residual we minimize combines the photometric, geometric and IMU residual.

6.3.1 IMU Factor

The IMU factor is computed by using the preintegration procedure of [52]. The IMU residual is composed of 3 terms - the residual of the relative translation between the frames, the relative velocity and the change in orientation.

$$r_{imu}(x, \Delta R_{ij}, \Delta v_{ij}, \Delta p_{ij}) = ||r_{\Delta p_{ij}}||^2_{\sum} + ||r_{\Delta v_{ij}}||^2_{\sum} + ||r_{\Delta R_{ij}}||^2_{\sum}$$

The variances for the 3 individual residual components is obtained from the expressions given in [51] and the variance for the combined term is computed via the standard means of combining variances for a linear combination of terms

$$\frac{1}{\sum_{imu}} = \frac{1}{\sum_{v}} + \frac{1}{\sum_{p}} + \frac{1}{\sum_{R}} \tag{6.1}$$

Under the assumption of Gaussian distributed noise, the IMU residual can be interpreted as the log-likelihood of the total residual

$$\{ (\mathcal{X}) = -\log p(\mathcal{I}|\mathcal{X}) = \sum_{k=1}^{n} r_{imu}$$

6.3.2 Semi-Dense photometric factor

We use a semi-dense photometric error term for computing the visual residuals for tracking. The semi-dense approach estimates the depth only at high-gradient pixels ensuring that there is enough texture for deriving reliable depth estimates.

$$r_{ph}(x, \Delta R_{ij}, \Delta v_{ij}, \Delta p_{ij}) = ||I_j \left(\pi \left(p_t^k\right)\right) - I_j \left(\pi \left(T_w^i T_w^j p_t^k\right)\right)||^2$$

The likelihood semi-dense factor is composed of the residuals

$$\{_{depth}\left(\mathcal{X}\right) = -\log p(\mathcal{D}) = \sum_{k=1}^{n} ||r_{ph}||^2$$

6.3.3 Sparse Geometric factor

Sparse features can greatly aid the tracking accuracy when highly textured, discriminative points are visible in the image. We therefore include the structure-less geometric constraints based on sparse feature points detected in the frames as in [51].

$$\{\left(\mathcal{X}\right) = ||y_k - h_\pi\left(\mathbf{x}_t^k, l_k\right)||^2$$

6.4 Depth Residuals

The depth-map predictions obtained from the network are typically not very accurate and cannot be used as is in the framework. As in [155], we use a depth map refinement procedure to improve the quality of the key-frame depth maps. As in [27] we optimize the photometric residuals of the depth map, but as opposed to regularizing the depth map using super-pixel fitted planes, we instead use the more-flexible predictions from the deep network.

$$\begin{aligned}\{_{\mathcal{D}}\left(\mathcal{X}\right) &= -\log p(\mathcal{D}) \\ &= \sum_{k=1}^{n} ||r_{ph}||^2\end{aligned}$$

The residual is computed in the same manner as the semi-dense factor,

$$r_{ph}(x, \Delta R_{ij}, \Delta v_{ij}, \Delta p_{ij}) = I_j\left(\pi\left(p_t^k\right)\right) - I_j\left(\pi\left(T_w^i T_w^j p_t^k\right)\right)$$

6.5 Key-Frame Selection

In this section, we detail our method for selecting keyframe iamges from the input stream. We propose to extract features based on the images and the odometry data to determine which images to use as keyframes. Keyframes satisfy three goals - firstly they need to ensure that the frames are of a **high quality** (i.e. not blurred or under exposed) secondly, they should produce an **even spatial sampling** of frames in the area being modelled which leads to better coverage and thirdly, **scene consistency** to ensure multiple views are captured of the same part of the scene. The last two goals are conflicting as an even spatial sampling precludes and vice-versa. Our optimal frame selection method involves first computing the factors and then performing a binary optimization to select the subset of frames that maximize their aggregate.

6.5.1 Selection Factors

We calculate the factors relating to the three requirements as follows:

Image quality: The SfM reconstruction process ideally requires input images that are crisp and clear, from which strong features can be extracted. Blurred images not only lead to a depletion of the number of features that are detected, but the blur can also drastically change the appearance of features, causing erroneous matches. In this way, a set of blurry images can easily lead to a reconstruction failure. Extracting highly-discriminative, repeatable features such as SIFT on the client device is very costly. To remain lightweight, we use a proxy for the number of features that can be extracted from an image. The proxy we use is the FAST corner detection method

[129]. By counting the number of FAST corners extracted in the image, we get a good proxy for the number of features that will be extracted at the server-end using a more computationally-expensive feature detection method. This feature is denoted

$$Q_{proxy}(i) \propto N_{FAST} \tag{6.2}$$

The quality factor should also account for the image blur. We use a vision-based factor, based on our lightweight proxy to identify blurry images. The number of corners, however cannot be applied straightforwardly to estimate the amount of blur, as the number of corners essentially measures sharpness which is a function of both the *content and the blurriness* of the image. To separate the appearance from this quantity we make use of the assumption that the appearance of the scene is constant over a short range of frames. In particular, we use the running average over the last 5 steps as a measure of the average number of features in the environment A, we then compute the relative sharpness value by comparing the instantaneous number of corners in an image to the moving average, calculated as

$$Q_{blur}(i) \propto \frac{N_{proxy}}{N_{mov.avg.}} \tag{6.3}$$

View consistency: This term we use accounts for the motion of the user. Ideally, more frames should be sampled during quick motions to ensure there is good overlap between frames. If the user is walking quickly or rotating, a large number of frames need to be sampled to satisfy requirement (2) to ensure that the features are re-observed for the reconstruction process. The *motion energy* of the user is used in our sampling algorithm to determine the number of frames that need to be sent in the initial batch. We accomplish this by calculating a motion-energy quantity which is proportional to *movement* of the user,

$$Q_{motion}(i) \propto v^2 + \omega^2 \tag{6.4}$$

The velocity, v, of the user is determined by the step counting and the angular rate, ω is read from the gyroscope.

Even spatial sampling: The final feature which we extract is the distance travelled by the user between two successive images. This is determined using the odometry derived from the images.

$$Q_{dist}(i,j) \propto \sum_{i}^{j} ||p_i - p_j|| \tag{6.5}$$

Once we have obtained the quality metric which allows us to rate the contribution of individual frames to the final reconstruction, we need to sample a set number of frames to send to the server. Selecting the K best frames amounts to maximizing the sum of the individual frame contributions as well as the pair-wise factors (i.e. the distance between frames). The purpose of the image selection is to select K images from the total set of N captured images while ensuring that the images passed to the reconstruction process satisfy the three requirements needed for high-fidelity reconstructions. We formulate the image selection process as an energy minimization problem with the following objective function using the factors defined in Eqn 6.2,6.3,6.4,6.5.

$$\begin{aligned} \underset{b\in\{0,1\}}{\text{minimize}} \quad & J(b) = \sum_i b_i Q(i) + \sum_{i,j} b_i b_j Q(i,j) \\ \text{subject to} \quad & \sum_i b_i \leq K \\ & 0 \leq b_i \leq 1. \end{aligned} \tag{6.6}$$

where $b_i \in \{0,1\}$ is a binary variable indicating that image i has been selected for use in reconstruction and will be transmitted to the server.

6.5.2 Selection Solver

The keyframe selection optimization problem in Eqn. 6.6 is a binary quadratic program (BIQP) which is known to be NP-hard [118]. Therefore, it is computationally intractable to solve quickly on a mobile device. We present our method based on the Projection onto Convex Sets (POCS) method [148] to solve it. The POCS methodology [148] solves constrained optimization problems by successively projecting onto a series of convex sets, the intersection of which constitutes the original constraints.

Algorithm 1 General form of gradient POCS solver for J(e)

1: **procedure** MINIMIZE J(P)
2: **while** $\delta > \epsilon_1$ **do**
3: $\tilde{\mathbf{b}}^{k+1} = \mathbf{b}^k - \lambda \frac{\Delta J(\mathbf{b}^k)}{||\Delta J(\mathbf{b}^k)||}$
4: $\mathbf{b} \leftarrow T(\tilde{\mathbf{b}}^{k+1})$
5: $\delta = ||\mathbf{b}^{k+1} - \mathbf{b}^k||$
6: **end while**
7: **end procedure**

As is evident from Eqn. 6.6, both constraint sets are convex and closed and the POCS method is readily applied in our case. Our POCS-based algorithm to solving this problem is outlined in Alg. 1. The core of the method is a gradient descent step where the gradient $\Delta J(\mathbf{b}^k)$ is the derivative of the relaxed objective equal to $-A\mathbf{b}^T - b$, where the matrix $A_{i,j} = Q(b_i, b_j)$ and the vector $b = Q(\mathbf{b}_i)$. As we show in our experiments, this algorithm typically converges in a very small number of iterations (usually around $50 - 100$) and in many cases obtains a solution which is very close to the global optimum.

A critical component of the algorithm is the projection operator $T(\cdot)$ which projects a candidate solution onto the constraint set. The projection function is implemented as an iteration which converges to a point in the constraint set. The approach we use to perform this projection is shown in Alg. 2. The iterative sequence of two successive projections, $\pi_a(x)$ and $\pi_b(x)$, related to the constraints of our problem are defined Alg. 2 lines 4,5, respectively.

Algorithm 2 Projection iteration for frame selection constraints

```
procedure T(b)
    while δ > ε_2 do
        b_h = b
        b ← π_a(b) with π_a(x) = min(max(x, 0), 1)
        b ← π_b(b) with π_b(x) = Kx/||x||
        δ_T = ||b_h − b||
    end while
end procedure
```

This simple yet powerful method of successive projections gives us an efficient means of obtaining a good solution to our frame subset selection problem, and comes close to the global optimal frame subset for our features.

6.6 An Integrated System

As we are interested in the effect of integrating our learned pose estimates into a standard SLAM system, we circumvent the computational burden of optimizing the residuals over each frame by performing the optimization in a two-stage process. First we optimize the full state vector over a window as in [92] to obtain a sequence of poses $\boldsymbol{\xi}_i$ relative to a starting frame. We then optimize these poses only at the keyframe locations. This process is used for both the MLE and MaP approaches.

6.6.1 MLE Estimation

The MLE optimization approach is the simplest SLAM system that can be formed using the learning-based pose estimates. It integrates *loop-closures* and and global *re-localization* information along with the relative motion predictions.

$$\operatorname{argmax} \log p(\mathbf{y}|\mathbf{x}) \tag{6.7}$$

The likelihood in this optimization includes the odometry summarized as relative

poses between keyframes as well as loop-closures to optimize a globally consistent map. The objective function is given by

$$E_S(\boldsymbol{\xi}) = \underbrace{\sum_{k=1}^{K-1} \mathbf{e}_{seq}^{k,k+1}(\boldsymbol{\xi}, \hat{\boldsymbol{\xi}})^\top \boldsymbol{\Omega}_{MDN}^{k,k+1} \mathbf{e}_{seq}^{k,k+1}(\boldsymbol{\xi}, \hat{\boldsymbol{\xi}})}_{\text{VIO prior}} + \underbrace{\sum_{(k,k')\in\mathcal{C}} (\boldsymbol{\xi} - \hat{\boldsymbol{\xi}})^\top \boldsymbol{\Omega}_{MDN}^{k,k+1} (\boldsymbol{\xi} - \hat{\boldsymbol{\xi}})}_{\text{re-localization prior}} \tag{6.8}$$

$$\mathbf{e}_{seq}^{k,k+1} := {}_{k+1}\boldsymbol{\xi}_k^{seq} \ominus \left(\boldsymbol{\xi}_{k+1} \ominus \boldsymbol{\xi}_k\right), \tag{6.9}$$

$$\mathbf{e}_{cls}^{k,k'} := {}_{k'}\boldsymbol{\xi}_k^{cls} \ominus \left(\boldsymbol{\xi}_{k'} \ominus \boldsymbol{\xi}_k\right), \tag{6.10}$$

6.6.2 M-a-P Estimation

Using our learned estimates and a traditional visual-odometry appraoch, visual-inertial localization and mapping can also be done using a maximum-a-posteriori (M-a-P) approach. In the M-a-P approach both a likelihood and prior is defined and the M-a-P estimate of the landmarks and the poses is found through an optimization:

$$\text{argmax} \log p(\mathbf{y}|\mathbf{x}) + \log p_\theta(\mathbf{x}) \tag{6.11}$$

$$E_S(\boldsymbol{\xi}) = \underbrace{\sum_{k=1}^{K-1} \mathbf{e}_{seq}^{k,k+1}(\boldsymbol{\xi})^\top \boldsymbol{\Omega}_{seq}^{k,k+1} \mathbf{e}_{seq}^{k,k+1}(\boldsymbol{\xi})}_{\text{visual-inertial odometry residuals}} + \underbrace{\sum_{(k,k')\in\mathcal{C}} \mathbf{e}_{cls}^{k,k'}(\boldsymbol{\xi})^\top \boldsymbol{\Omega}_{cls}^{k,k'} \mathbf{e}_{cls}^{k,k'}(\boldsymbol{\xi})}_{\text{loop closure residuals}} \tag{6.12}$$

$$\underbrace{\sum_{k=1}^{K-1} \mathbf{e}_{seq}^{k,k+1}(\boldsymbol{\xi}, \hat{\boldsymbol{\xi}})^\top \boldsymbol{\Omega}_{MDN}^{k,k+1} \mathbf{e}_{seq}^{k,k+1}(\boldsymbol{\xi}, \hat{\boldsymbol{\xi}})}_{\text{VIO prior}} + \underbrace{\sum_{(k,k')\in\mathcal{C}} (\boldsymbol{\xi} - \hat{\boldsymbol{\xi}})^\top \boldsymbol{\Omega}_{MDN}^{k,k+1} (\boldsymbol{\xi} - \hat{\boldsymbol{\xi}})}_{\text{re-localization prior}} \tag{6.13}$$

Where $p(\mathbf{y}|\mathbf{x})$ is the likelihood of the input data given the current state estimate and $p_\theta(\mathbf{x})$ is the parametric prior representing knowledge originating from elsewhere.

In standard visual-inertial localization and mapping systems, either a uniform and uninformative prior is used [53] or a simple assumption is made, for example, the constant velocity model.

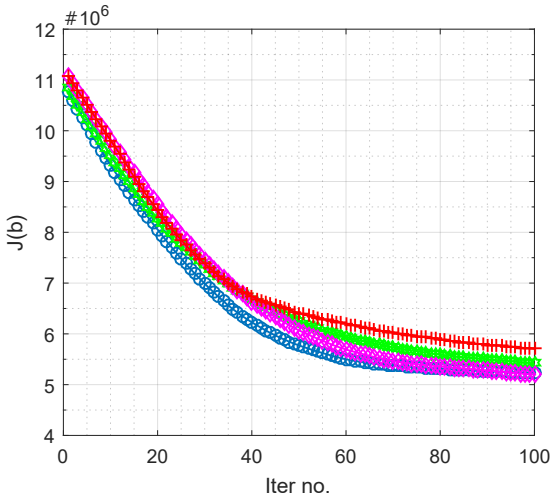
The M-a-P estimate for the dense depth map is obtained by interpreting the prediction of the deep network as a prior of a variational framework

$$E(\mathbf{d}) = \int_{\Sigma} \underbrace{f(\mathbf{u}, \mathbf{d}(\mathbf{u}))}_{\text{photometric error}} + \lambda_1 \underbrace{||\mathbf{d}(\mathbf{u}) - \text{UNet}(\mathbf{u})||}_{\text{deep prior}} + \lambda_2 \underbrace{g(\mathbf{u})||\nabla \mathbf{d}(\mathbf{u})||}_{\text{regularization}} \tag{6.14}$$

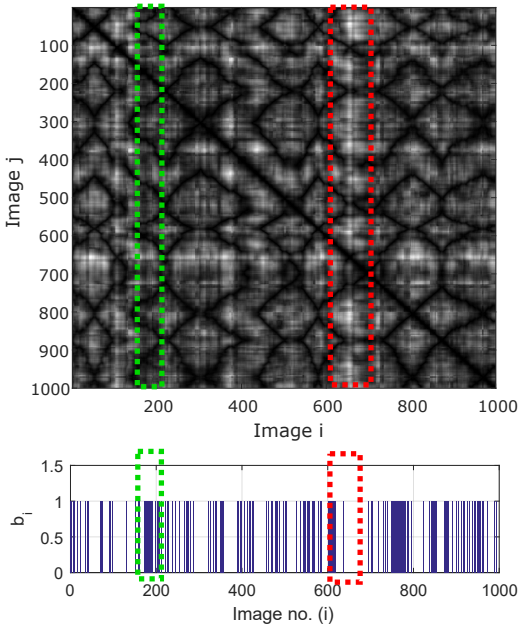
As in [27, 116], optimization is carried out using a primal-dual method for TV-L1 denoising.

6.7 Experiments

In this section we evaluate the performance of the proposed visual-inertial SLAM system with deep priors. We compare our approach to existing visual-inertial systems, namely OK-VIS [92], VI-SLAM [81] and a vision-only approach LSD-SLAM [44]. We also compare our system to the most closely related method, CNN-SLAM [155] which augments LSD-SLAM with depth maps predicted from a deep network. We evaluate the proposed keyframe selection method and as well as the effect that the deep priors have on the accuracy of the trajectories estimated by the system. The initial calibration parameters we assume are shown in Table 3.1.



(a) Convergence of the binary optimization algorithm on the test data. The global optimum is $\approx 4.5 \times 10^6$.



(b) Result of the optimization with a cost matrix derived from real data.

Figure 6.1: Performance of the keyframe selection optimization method.

6.7.1 Keyframe selection

We first test our efficient keyframe selection algorithm on real-world data from the office environment. We select a subset of 1000 images for which we can compute a ground-truth optimum which we find using MATLAB.

The operation of the image selection algorithm on real-world data for selecting $K = 200$ images out of a set of $N = 1000$ is illustrated in Fig. 6.1b. The solution is shown in the lower subplot. By comparing the two figures we can gain insight into the operation of the image selection algorithm and verify that it operates according to intuition. For example, the green and red blocks show two segments of images where many and few images, respectively, were selected. By looking at the potential matrix in the top subplot we see the region where many images were selected the compatibility function between the images was low. This may represent a region where there was, for example, good displacement with little heading change. In the

region marked by the red square, on the other hand, very few images were selected. This region consisted, for example, of only rotational motion and no walking.

To test the sensitivity of the optimization algorithm to the starting point, we run the optimization multiple times on the same test data. The results of this experiment are plotted in Fig. 6.1a. From these results, we observe two interesting trends. Firstly, the randomized starting point does not greatly affect the initial value of the objective function. This indicates that a randomly selected image is not a very effective strategy, as expected. Secondly, the results indicate that the proposed optimization algorithm is not very sensitive to the initial conditions and converges well in the case of real-world data.

6.7.2 Trajectory accuracy

In this section we evaluate the pose accuracy obtained by our visual-inertial SLAM system with online optimization. We evaluate its performance on the KITTI and EuRoC datasets. For the priors, we use the same trained ego-motion network models as used for the results in Chapter 5.

Table 6.1: Accuracy on the EuRoC dataset

Sequence	MH01	MH02	MH03	MH04	MH05	VR11	VR12	VR13
OK-VIS	0.34	0.36	0.3	0.48	0.47	0.12	0.16	0.24
VI SLAM	0.25	0.18	0.21	0.3	0.35	0.11	0.13	0.2
OK-VIS, stereo	0.23	0.15	0.23	0.32	0.36	0.04	0.08	0.13
LSD-SLAM	n/a	n/a	n/a	n/a	n/a	0.07	0.07	0.09
Proposed	0.28	0.11	0.20	0.40	0.39	0.11	0.17	0.18

Table 6.1 reports the results on the EuRoC dataset for our integrated system, along with OK-VIS and LSD-SLAM. The monocular depth prediction on the EuRoC dataset is rather inaccurate as the lens setup varies greatly (in terms of focal length, camera distortion and environment appearance) from the datasets on which the monocular depth prediction network was trained. However, even without accurate

dense depth prediction, we found that the performance of the proposed system still performed well (mainly due to the incorporation of sparse features) and outperformed state-of-the-art sparse monocular approaches. In fact, the system with learned priors is even competitive to a stereo vision-inertial approach as evident from Table 6.1.

Table 6.2: Accuracies obtained on KITTI.

		VISO2-Stereo		Prior (Chap 5)		Proposed	
Seq	Frms	Rot	Trans	Rot	Trans	Rot	Trans
		(deg/m)	(%)	(deg/m)	(%)	(deg/m)	(%)
0	4540	0.0109	2.32	0.0037	3.2877	0.0026	2.34
2	4660	0.0074	2.01	0.0025	3.0650	0.0015	2.98
3	800	0.0107	2.32	0.0052	2.8490	0.0044	2.64
4	270	0.0081	0.99	0.0031	3.2741	0.0021	2.41
5	2760	0.0098	1.78	0.0061	2.6577	0.0055	1.98
6	1100	0.0072	1.17	0.0032	2.9796	0.0030	1.64
8	4070	0.0104	2.35	0.0077	2.9517	0.0080	1.8
9	1590	0.0094	2.36	0.0045	3.0638	0.0040	2.56
10	1200	0.0086	2.12	0.0048	2.8273	0.0031	2.846
Avg		0.0094		2.06		0.0045	

The results on the KITTI dataset are shown in Table 6.2. The table shows the advantage of the full SLAM system over the raw motion priors, as introduced in Chapter 5. This gain is rather significant in most cases, for example, in Sequence 8 a 63% increase in accuracy is obtained. Again, the proposed system nears the performance of the VISO-2 Stereo visual-odometry approach. In Table 6.3 we provide an evaluation of our approach on the TUM RGBD dataset [150] and provide a comparison to the CNN-SLAM [155] method. Our system outperforms CNN-SLAM and LSD SLAM on most of the sequences, which can be attributed to our motion priors as well as our utilisation of both sparse and dense features. The relative errors (as evaluated using the relative pose error approach described in [150]) over 1m window is shown in Figure 6.4. In the figure, the re-localizations against the current map can be seen $T = 23s$, $T = 26s$ and $T = 27s$. Although our proposed approach loses tracking more often than the RGB-D based method [28], it provides accuracy that is comparable to

the RGBD appraoch without requiring depth input.

Table 6.3: Comparison of the proposed method to CNN-SLAM on the TUM RGBD dataset (no depth data is used to perform the tracking).

	Absolute Trajectory Error					
	CNN-SLAM	LSD-BS	LSD	ORB	Laina	Proposed
ICL/office0	0.266	0.587	0.528	0.430	0.337	0.250
ICL/office1	0.157	0.790	0.768	0.780	0.218	0.135
ICL/office2	0.213	0.172	0.794	0.860	0.50	0.444
ICL/living0	0.196	0.894	0.516	0.493	0.230	0.180
ICL/living1	0.059	0.540	0.480	0.129	0.060	0.041
ICL/living2	0.323	0.211	0.667	0.663	0.380	0.201
TUM/seq1	0.542	1.717	1.826	1.206	0.809	0.304
TUM/seq2	0.243	0.106	0.436	0.495	1.337	0.138
TUM/seq3	0.214	0.037	0.937	0.733	0.724	0.136
Avg.	0.246	0.562	0.772	0.643	0.512	0.203

Finally in Figure 6.2 we show a qualitative comparison to the CNN-SLAM method on the challenging *rgbd_dataset_freiburg1_xyz* sequence from the TUM dataset which contains mostly rotational motion. The benefit our proposed voxel-based refinement of the dense depth maps can be seen from the final point which is smoother and more dense compared to CNN-SLAM [155]. In Figure 6.3 we show a larger scale reconstruction using our method with only monocualar input. Most of the mapping error comes from the depth prediction which is relatively inaccurate in areas such as near the window on the right of the room.

As our system relies on a learned network for its prior, generalization is a significant consideration. However, as the features learned by our CNN are geometric in nature (i.e. they are related to the optical flow rather than appearance), it can generalize fairly well to new environments without retraining. For the integrated system, generalization issues are further mitigated as the pose accuracy is boosted by the optimzation which requires no environment-specific training.

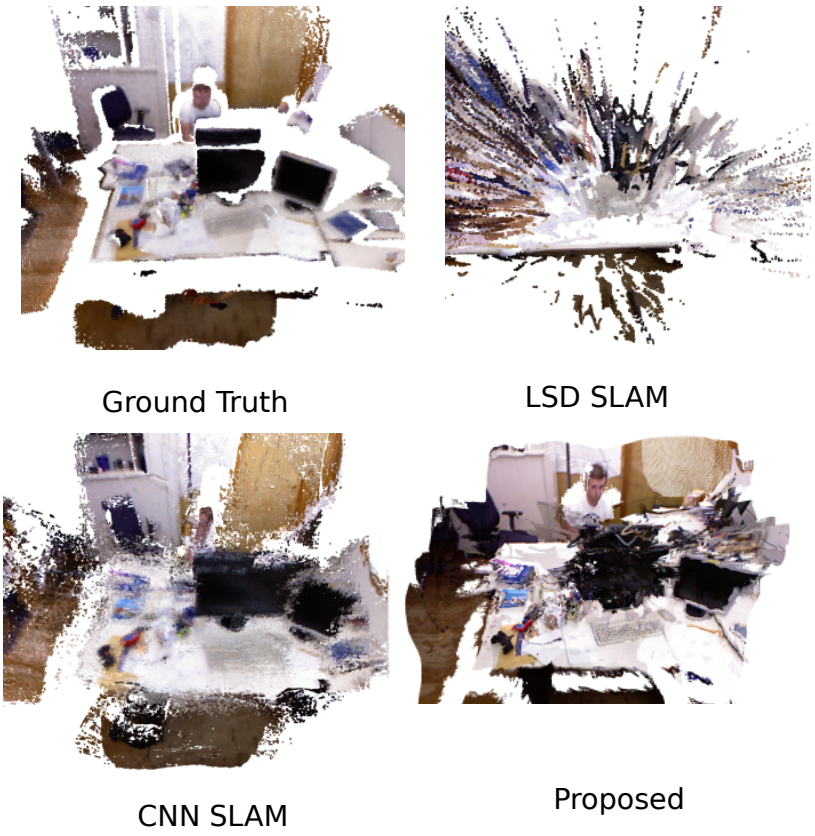


Figure 6.2: Comparison between the proposed monocular reconstruction system with deep motion and depth priors, LSD-SLAM and CNN-SLAM on rgbd_dataset_freiburg1_xyz.

Figure 6.3: Reconstruction of rgbd_dataset_freiburg1_room using our method (no depth information is used).

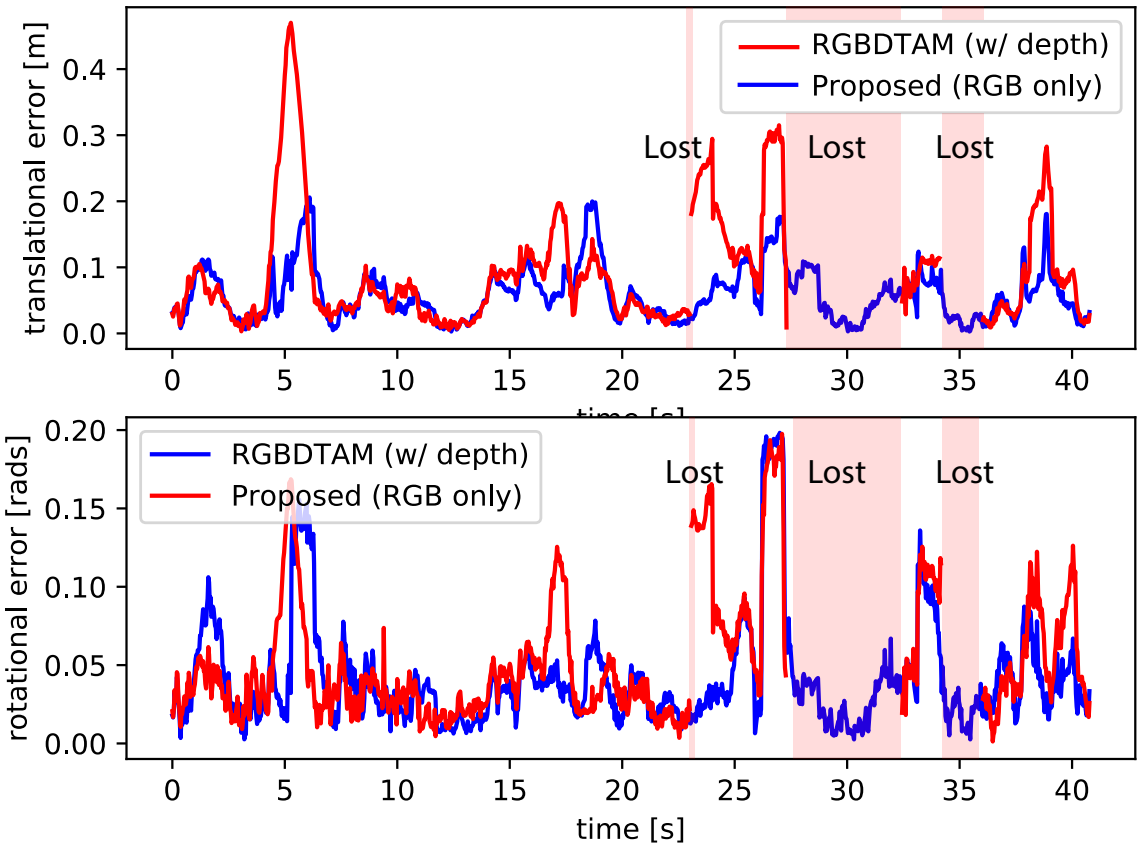


Figure 6.4: Errors on the rgbd_dataset_freiburg1_room sequence from the TUM RGBD dataset.

6.8 Conclusion

In this chapter we analysed the benefit that deep priors can have on a traditional visual-inertial SLAM system incorporating loop-closure, re-localization and online optimization through bundle adjustment. We have proposed a system which tightly integrates the learned motion and location priors. Our system builds on the existing dense visual-inertial approach [108] and integrates the learned priors proposed in the preceding chapters of this thesis.

Chapter 7

Efficient Localization

In this chapter, we propose an Efficient Multi-modal Localisation (EM-Loc) system, which leverages multiple modalities to make precise vision-based localisation on resource-constrained platforms more feasible. We exploit the side-channel information and along with an estimated trajectory predict which features will be visible. This intelligent guidance is achieved through a sequential Monte-Carlo process which estimates the posterior distribution of the current location over the nodes which ensures the 3D pose estimation is carried out using only the currently visible features. We show how this dramatically reduces computational time for 6-DoF localisation and achieves high accuracy.

The key contributions of this chapter are:

- We present EM-LOC, an accurate localisation system that can be used across a range of platforms
- We propose a localisation policy which exploits side-channel information (WiFi, magnetic) to cut down vision-based localisation time (thus enabling real-time 6-DoF visual localisation)
- We evaluate these contributions on data gathered from a large museum building

The remainder of the chapter is organised as follows: Sec. 7.5 focuses on the two aspects of the EM-LOC system - graph construction and localisation respectively. Sec. 7.6 presents an evaluation of the proposed system and Sec. 7.7 concludes the chapter and highlights ideas for future work.

7.1 Related publications

The publications arising from the work in this chapter include

- **Clark, R.**, Wen, H., Wang, S., Trigoni. N., Markham, A. Increasing the Efficiency of Visual Localization using Multi-Modal Sensing. In Proceedings of the IEEE RAS International Conference on Humanoid Robotics (Humanoids), 2016.

- Wang, S., Wen, H., **Clark, R.**, Trigoni. N. Large-scale Keyframe-based Localization using Geomagnetic Field and Motion Pattern. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), 2016.

- **Clark, R.**, Markham A., Trigoni. N. Abstract: Towards Robust Vision-based Indoor Localization. In Proceedings of the IEEE International Conference on Information Processing in Sensor Networks (IPSN), 2015.

7.2 Related work

Multi-modal localisation: The promise of exploiting multiple modalities for aiding vision-based localisation has been demonstrated in a variety of ways, but not directly for reducing visual processing time. For example, the W-RGBD system [75] uses WiFi in combination with RGBD images to perform localisation using an architectural floorplan. They model the WiFi signal strength across the floorplan using a Gaussian process and through the use of this model intelligently initialize the particles for

Monte Carlo localisation (MCL). They show that this improves the convergence rate and final accuracy of the position estimate. Their goal differs from ours; while they are mainly concerned with using WiFi to increase the convergence rate and global accuracy of MCL we are interested in using the side-channel information to guide the expensive 2D-3D feature matching to increase the efficiency of the RANSAC-based pose computation and thus reduce the computational time required on resource-constrained platforms.

7.3 Proposed System

Our system consists of a two-tier architecture, as shown in Figure 7.1. The back-end graph assembler is responsible for creating the localisation graph, while the front-end runs on the target platform and performs on-line localisation. The back-end graph assembler receives as input the data collected from mappers during the mapping session of the system which are uploaded to the cloud where the server is implemented. The graph assembler constructs the localization graph by merging data traces from different mappers. The localiser carries out state estimation to position the humanoid with respect to the assembled localisation graph.

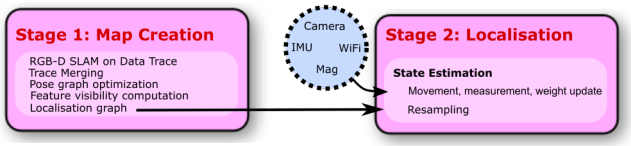


Figure 7.1: EM-LOC consists of a two-tier architecture. The localisation graph is created and updated in the cloud, while the localisation module receives the localisation graph and performs on-line state estimation.

7.4 Stage 1: Map creation

We use three types of information in our map creation phase - RGB data from which sparse SIFT features [97] are extracted and associated with 3D points (these are needed to perform the 6-DoF localization), depth frames (which are used to determine the visibility of the sparse feature points from nodes) and the multi-modal side-channel information. Using this data a map of 3D points associated with SURF features and the relative translation between frames is created using an RGB-D SLAM algorithm [42] [1] The localization graph in our case is a set of nodes, $n_1, n_2, \ldots, n_N$ (each frame is considered as a node) which are joined by a set of directed edges $e_{i-1,i}$ that link the nodes. Each edge contains odometry data consisting of the displacement and orientation (computed by the RGB-D SLAM) which describes the metric relation between the nodes. Each edge also contains the side-channel information which includes the distribution of WiFi RSSI values along that edge, geo-magnetic distortion measurements and appearance-based visual features.

The structure of the side-channel features is as follows:

WiFi: The readings are composed of the received signal strength, s, for each access point observed in the environment. As described in [143], the RSS of typical radio-based measurements does not change very rapidly with typical humanoid/human mobility and thus the distribution is collected over a finite duration of time which is used as the feature, $E_i^{wifi} = [\mathrm{s}_1, \mathrm{s}_2, \cdots \mathrm{s}_n]$

Magnetic: Similarly, we capture the magnetic field strength data over a short time segment. The magnetic data consists of the magnitude, m, of the geo-magnetic field calculated from the 3 components in the x, y and z axes, $E_i^{mag} = [m_1, m_2, \ldots, m_n]$

Image: We further capture images along each experience and use appearance-based features derived from these images as additional side-channel information. The full

[1] If a depth camera is not available, this process could also be carried out using an image-based Structure-from-Motion process followed by a dense multi-view stereo reconstruction.

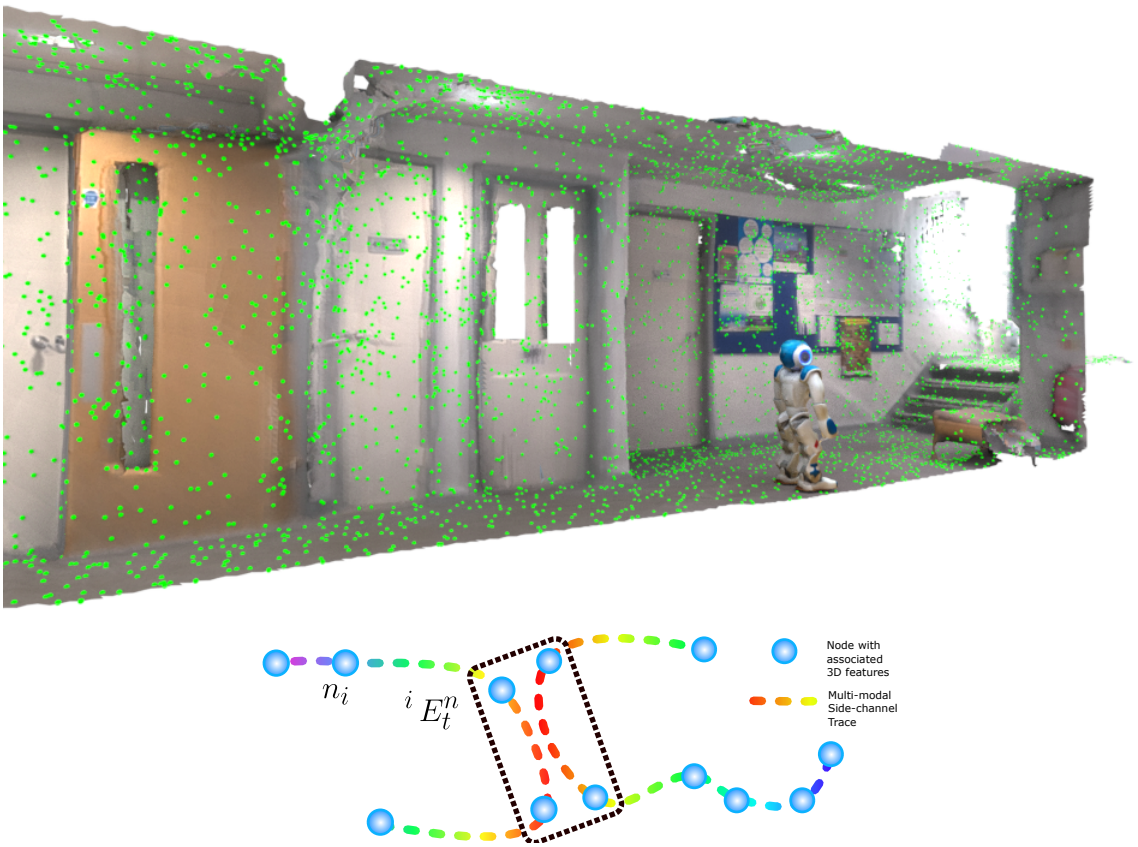


Figure 7.2: Illustration of a 3D model for localization with millions of candidate feature points (green dots).In this chapter, we propose EM-Loc (Efficient Multi-modal Localization) that integrates these features in a localization graph and interweaves it with additional (side-channel) sensory data. Our localiser uses this side-channel information to predict which features will give successful visual localization. This cuts down the matching time and enables real-time and fine-grained localization over vast areas.

images themselves are not stored, rather we detect and compute the distinctive visual features which are present in the image. Each feature is then quantized into BovW vector, where the word vocabulary, $w_1, w_2, \cdots, w_n$ has been precomputed using k-means clustering. The final representation of the image consists of a histogram of the word occurrence counts $E_i^{img} = [h_1, h_2, \cdots, h_n]$ As the quantization incurs additional cost, the inclusion of appearance-based features is entirely optional in our system, and if these features are used the vocabulary is kept very small.

The graph consists of data traces, which may be collected during different times, allowing the map to be readily extended. Many disjoint data traces may be generated by mappers traversing an area. The localisation graph thus consists of short segments, along with their nodes and edges, which need to be linked together at certain co-location points. The graph assembly process harnesses the side-channel information to establish these co-location links across multiple data traces. Given a trace denoted as E_i^n where n is the modality, i is the trace index and t is the timestamp at which the data was recorded, a co-location link established across multiple traces when

$$\exists t_2 \text{ s.t. } \max P({}^iE_{t_1}^n | {}^iE_{t_2}^n) > \tau. \tag{7.1}$$

This means is that for each data frame of a certain modality that is assigned to a place, some other modality, already belonging to that place shares a strong similarity with its partner in the current trace. After the co-location links have been established the entire localisation graph is optimized using a robust pose graph optimization through the GT-SAM library [34] with the Vertigo extension [151]. The graph construction process is carried out off-line and also updated using data collected during additional mapping sessions. The end result of this process is a consistent representation of the world in the form of a linked multi-modal localization graph.

Once the graph has been created and optimized it consists of a large number of nodes each associated with a small number of feature points which represent those that were visible in the frame during the mapping process. However, once the consistant map has been created, many feature points may be visible from a node which is not taken into account by the initial mapping process. We therefore utilise a post-processing step to determine the true point visibility for each node. This is achieved by raycasting from the node to the 3D location of each SIFT feature stored in the

map and checking for ray intersections with depth frames registered to their world locations. If no hit occurs occurs, the SIFT feature is visible from that node and is added to the node's feature-set. This process is illustrated in Fig. 7.3. Each node is associated with a set of 3D features which comprises those features that are visible from the node's location.

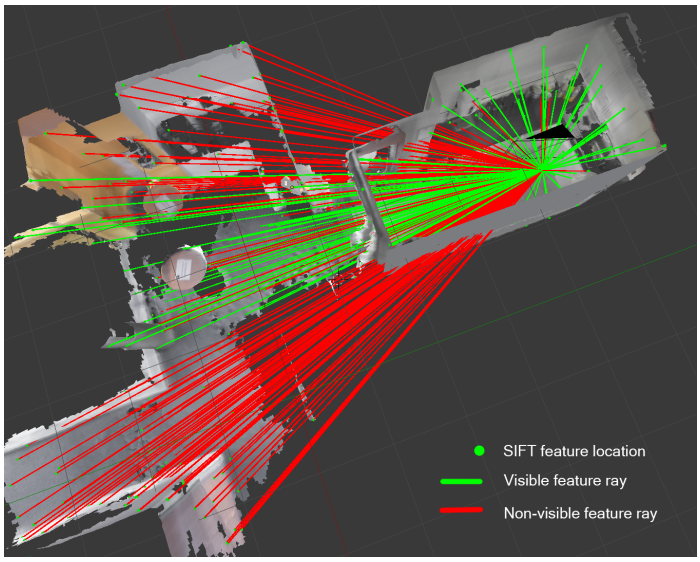


Figure 7.3: In order to determine the visbility of features from each node location, we make use of the depth frames. The visbility is computed by ray-casting from the node center to the stored 3D location of the feature.

7.5 Stage 2: Localisation

In this section we describe our localiser which runs on the humanoid with the objective of providing efficient 6-DoF pose estimates. We adopt a sequential Bayesian approach to perform localisation in the established localisation graph.

$$p(n_i, \mathbf{X}_i, \Theta_i | z_{i:t}) = p(\mathbf{X}_i, \Theta_i | n_i, z_{i:t}) p(n_i | z_{i:t})$$

Where n_i are the nodes of the localisation graph as defined before, Θ_i is the orientation (yaw, pitch, roll) of the device, and $\mathbf{X}_i$ is the location of the humanoid *with respect*

to the current node n_i. The pose estimates relative to a node are independent of the node, the state estimation problem reduces to

$$p(n_i, \mathbf{X}_i, \Theta_i | z_{i:t}) = p(\mathbf{X}_i, \Theta_i | z_{i:t}) p(n_i | z_{i:t})$$

In order to perform localisation, we use a particle filtering approach. This enables us to employ a Rao-Blackwellized approach to performing the localisation on the localisation graph where the term $p(n_i|z_{i:t})$ keeps track of the global localisation of the humanoid in the localisation graph, while the term $p(\mathbf{X}_i, \Theta_i|z_{i:t})$ ensures that the local 6D pose computations remain consistent. In order to ensure that the pose computation remains lightweight, we make the assumption that the local pose estimate produces a uni-modal Gaussian. Our particle-filter is initialized and updated as follows.

7.5.1 State and initialization:

The particle structure is split into two components which consists of an estimate of the humanoid's current location in the localisation graph represented by the node n_i and $(x_i, y_i, z_i, \theta_i)$ which is the 6-DoF pose of the humanoid relative to node n_i.

$$\text{Particle: } X_i = [\{n_i\},] \, [\{\mu_{\mathbf{X}_i,\Theta_i} \mathbf{\Sigma}_{\mathbf{X}_i,\Theta_i}\}]$$

We uniformly initialize the particles over the entire localisation graph.

7.5.2 Measurement

In order to utilise the incoming sensory measurements for localisation in our localisation graph, we need some means of relating the measurements to the data stored

in our graph. This is achieved by utilising likelihood functions for each modality as a function of the system state. In particular, we use the following likelihood functions:
RSSI: For RSSI data, the Kullback-Liebler divergence, D_{KL}, is used as the metric to compare measurements coming from individual access points.

$$p(E_i^{wifi}|Ei_j^{wifi}) \propto e^{-D_{KL}\left(E_i^{wifi},E_j^{wifi}\right)}$$

Magnetic: The magnetic features consist of a series of geo-magnetic measurements strung together in a series. As the humanoid may have been moving at different speeds between the stored and query feature, we use dynamic time warping (DTW) as a measure of their similarity

$$p(E_i^{mag}|E_j^{mag}) \propto e^{-DTW(E_i^{mag},E_j^{mag})}.$$

Image: To compute the likelihood of a new image given one in the stored experience we use use the FAB-MAP approach [32]. FAB-MAP takes as input the image's BoW vector and produces a normalized likelihood measure for each stored location, $p(E_i^{img}|E_j^{img})$.
We compute the weight by assuming that each of the side-channel likelihoods is independent. The importance weights, w_t^i, for the particles are thus calculated as the product of the individual terms

$$w_t^i = p(E_i^{wifi}|Ei_j^{wifi})p(E_i^{mag}|E_j^{mag})p(E_i^{img}|E_j^{img}).$$

7.5.3 Odometry

For the process update of our sequential Bayesian filter, we use odometric measurements. Many humanoid odometry estimation schemes rely on incorporating domain constraints for a specific application, along with a complex dynamics model [162] and state-estimation methods. In this chapter, however, we are interested in the case where the IMU is not firmly attached to the body of the subject being tracked with the goal of allowing our approach to be transferred seamlessly from a humanoid to human (eg. human wearing a pair of smart-glasses). In this case such strict and clear-cut assumptions does not exist and thus, rather than relying on a state-space approach and encoder data, most methods extract features from the IMU traces and use these these for odometry estimation in what is know as pedestrian dead reckoning (PDR) [57]. PDR performs 3 main operations; step detection, step-length estimation and heading calculation. By coupling the stride length estimates and step event detections derived from these features, the PDR is able to roughly estimate the relative position of the subject. Step events are detected as maximum of the z-axis accelerometer data and the heading is tracking using the the gyroscope and magnetometer [85]. We use a simple step-frequency which has been shown to work well in practice [127]. This model relates the step length to the step frequency, $l_s = \alpha f + b$, where f is the step frequency and α and b are constants as described in [127].

7.5.4 Pose Estimation using 3D features

6-DoF pose estimation using 3D features is performed using the EPnP algorithm [90]. Using a set of 2D features detected in an image at locations, u_i, v_i matched to 3D features x_i^N, y_i^N, z_i^N stored relative to the coordinate system of node n. The pose, $\mathbf{X}, \theta$, of the the camera which recorded the image relative to the world coordinate system, is computed using a PnP solver. We use the standard EPnP solver wrapped in a RANSAC loop as implemented in OpenCV [14].

7.5.5 EKF Process and Measurement Update

Unlike the side-channel information which gives inexpensive, but noisy and outlier-prone estimates of the current location, the 6-DoF pose computation gives outlier-free measurements. An internal EKF is used to keep track of the pose mean $\mu_{\mathbf{X}_i,\Theta_i}$ and the covariance $\mathbf{\Sigma}_{\mathbf{X}_i,\Theta_i}$, *relative* to a particular node. The process update of the EKF is carried out by using odometric measurements.

In particular, process update is carried out as follows

$$x_n^t = x_n^t + l_s \times cos(\theta_\phi)$$
$$y_n^t = y_n^t + l_s \times sin(\theta_\phi)$$

Where the step lengths l_s and step detections are obtained from the PDR. For the EKF measurement, a 6-DoF pose computation is done using the 3D features, this yields an observation $z = \hat{x}_i^N, \hat{y}_i^N, \hat{z}_i^N, \Theta_i$ relative to the 3D node, n_i, to which those features belong. Using this measurement, the Kalman gain is calculated using the standard gain equations [69] and the mean and covariance updated using this gain. We concurrently perform the measurement update for the weights of each particle sharing the same node n_i, which significantly reduces the computational effort.

7.5.6 Update and resampling

The weights consider only the likelihoods of the particles at the discretised node-level. However, our particles contain more information i.e. in the form of a continuous 6-DoF pose estimate (maintained by the EKF). As the dead reckoning and 6-DoF pose estimate may lead to the particle deviating from the current nearest node and coming closer to another one, we further adjust the weight of the particles such that those

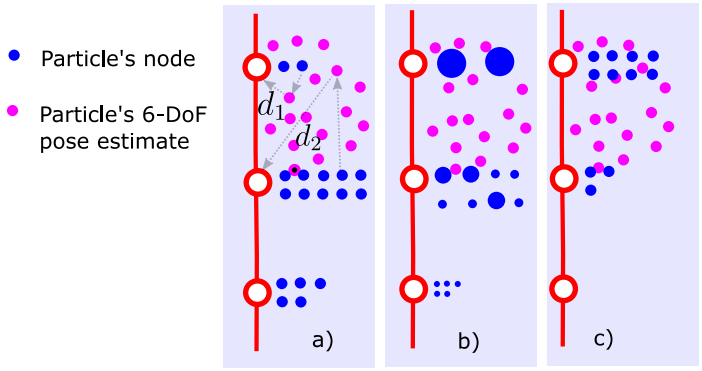


Figure 7.4: a) Before particle re-weighting many particles contain 6-DoF pose estimates which are far from their current graph node. b) After weight modification, particles with with nodes far from their 6-Dof estimates are given a lower weight. c) After re-sampling more particles have closely-related node and 6-DoF pose estimates.

closer to the predicted pose have a greater weight.

$$w_k^{i'} = w_k^{i'} e^{-d}.$$

The quantity $d = \sqrt{(x_k^{i'})^2 + (y_k^{i'})^2 + (z_k^{i'})^2}$ represents the distance of the particle's 6-DoF pose estimate to the location its node i'. The transformation to a particular node's coordinate system is done using the information stored along the edges in the graph. This weight adjustment is illustrated in Fig. 7.4.

7.5.7 Localisation policy

A major bottleneck in using full 3D localization for the purpose of mobile device localisation is the severe limitation which the computational constraints of the mobile device places on the number of 3D features which can be localized against while still allowing for real-time operation. To address this, we use a localisation policy which performs the expensive 2D-3D matching only in nodes where it is essential, guided by the current distribution of particles on the graph.

For reliable localisation, we would like to ensure that 2 criteria are met; 1) the

particle distribution should faithfully represent the posterior distribution of the humanoid's location on the graph and 2) we only want to try to localize on a limited number of nodes and these should have a high probability of being the humanoid's actual location.

In order to satisfy 1) we use a metric based on the importance weights obtained from the previous time instant, w^i_{t-1}. These weights are used to calculate the "effective number of particles" $N_{eff} = \frac{1}{\sum \hat{\omega^i}^2_{t-1}}$, which is a metric that gauges how well the particles represent the posterior [39].

The localisation policy then uses this metric to determine when localisation should be attempted. If the effective number of particles is high enough, the localiser attempts to localize in the k most likely nodes. Typically, the 3D matching only has to be done on 1 or 2 nodes and each of these frames *naturally* contains only a fraction of the most relevant 3D features needed for localisation. The full localisation algorithm is detailed in Alg. 3.

Algorithm 3 Localizer

Require: localisation graph G, Particles X_i

```
1: function LOCALIZE(G,X_i)
2:     Calculate N_eff
3:     for m = 1 : k do
4:         if N_eff > τ_eff then
5:             Select m^th most common node
6:             Pose computation using m's 3D features
7:         end if
8:     end for
9:     return X_t, θ_t                          ▷ Proceed with EKF update
10: end function
```

The quantity k is a very important parameter as it determines the number of nodes in which localisation is attempted. It therefore directly affects the number of feature matches that are carried out during the 6-Dof pose estimate. By adjusting this quantity, the number of feature matching can be controlled as required and thus

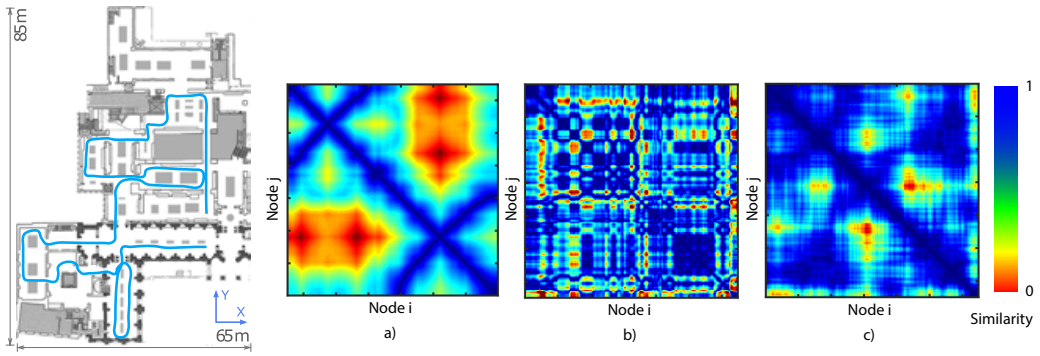


Figure 7.5: Visualization of the magnetic and WiFi similarities for the data used in the museum experiment. a) shows the ground-truth metric distance between the nodes b) shows the magnetic signal similarity and c) the WiFi similarity

allows the localisation time to be scaled according to the requirements of the platform.

7.6 Experiments

In order to adaquately evaluate EM-LOC, we gauge its performance along three performance axes. The first is the **localisation time**. As the 6-DoF localisation is the most computationally expensive component which we seek to minimize, we define as the localisation time as the time it takes from receiving the current visual frame from the camera to the time that the 6-DoF pose for the frame is determined by the RANSAC-based pose computation. The second factor is the **localisation efficiency**. The localisation efficiency is defined as the number of successfully localised divided by the total number of images for which localisation was attempted. This is important as the RANSAC-based pose computation may not localise a 6-DoF pose if it does not find enough inliers. In our case we use 8 inliers as the threshold for successful localisation. The final aspect we evaluate is the **localisation accuracy** which is simply the difference between the system's current estimate of the angle and orientation and the ground truth pose.

For fair comparison, we compare to two existing methods in terms of both accuracy

and processing time. The first system, which we use as a baseline for comparison is the EBN framework [23]. The EBN method stores frames along with 3D feature points against which incoming images can be localised. We consider two variants of the framework. The first carries out an exhaustive matching against 3D points belonging to all available nodes stored in the map - in order to vary the number of nodes that are used in the localisation, we randomly select k nodes from the graph (we label this method A1). In the second method for comparison, we make use of the side-channel data to rank the nodes which are selected (A2). The ranking is computed by using the likelihoods of Sec. 7.5.2 as features and using the independence assumption - ranking the nodes as the product of the features in the same way as the weights of the particles are calculated in EM-LOC. Our third method for comparison is based on Travi-Navi [168], which is a state-of-the art teach-and- repeat pedestrian bi-pedal navigation system from the mobile computing literature. Travi-Navi simply uses an SVM to produce an estimate of the current location from the WiFi, magnetic and appearance-based visual data. For our purposes, we use the SVM to rank the nodes on which the 3D pose estimation is performed (A3).

Our evaluation data was gathered from a large museum area. The area was chosen for numerous reasons; firstly, its size ($\approx 5000m^2$) allows for thorough evaluation of our approach; secondly it is representative of a practical implementation setting of our system - a popular tourist attraction and public site with much commercial interest for a viable localisation service. Furthermore, the museum interior is a very complex space - filled with vast open rooms, narrow corridors and dim-lighting which makes it a challenge for vision-based localisation methods. In order to evaluate such a large area and considering the locomotion speed of current humanoid robots, the data for our experiments were collected by two humans equipped with a pair of Google Glasses. This is reasonable as our odometry does not rely on modelling intricate dynamics. For offline procedure of map creation, RGB-D frames were collected using a Project

Tango device and run through an RGB-D Slam algorithm [42]. The same procedure was used to collect ground-truth locations for the online localisation tests. We stress, however, that all the reported online localisation results are obtained using the data from Google Glass. We test at a total of 167 test locations across the museum. Our experiments were run on an Intel Core i5-2467M @ 1.60GHz, which is similar in performance to that of the Nao v5.

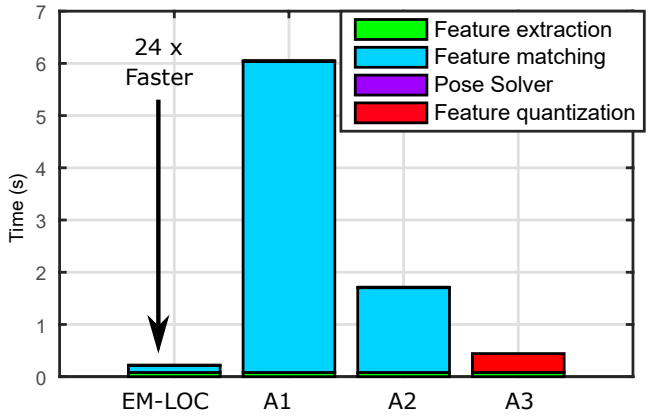


Figure 7.6: Comparison between the processing times for EM-LOC system, and comparison methods A1, A2 and A3 (see text)

The error distribution for EM-LOC, A2 and A3 at 3 values of k is reported in the left subplots of Fig. 7.9 (A1 is not plotted in the graph as the random node decimation performs significantly worse than the other methods at these low k values) and the CDF in Fig. 7.8. The EM-LOC system clearly outperform methods A2 and A3 in terms of localisation accuracy. This can be explained through two observations. Firstly, EM-LOC, has the ability to better propose matching candidates for the localisation. This means that nodes where localisation may be possible, but of low accuracy (such as distant nodes) are automatically excluded. Secondly, by leveraging multi-modal data, EM-LOC is able to resolve visual ambiguities such as corridors of similar appearance which are common very common in indoor environments and in many cases cannot be resolved using visual data alone. These two factors contribute

to the 6-DoF pose estimation in EM-LOC producing fewer outliers and more accurate pose estimates.

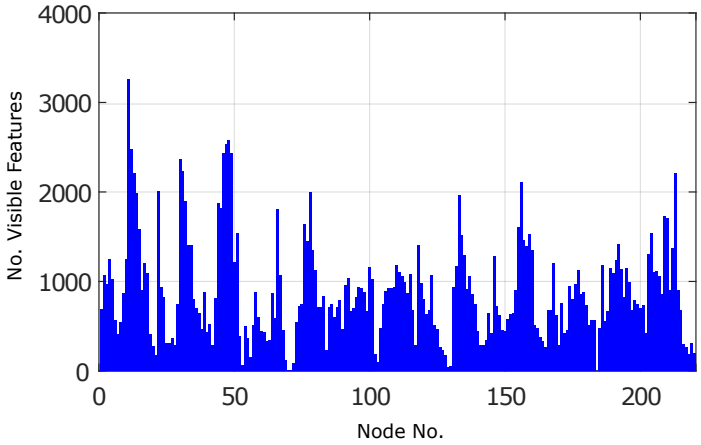


Figure 7.7: Number of features visible for each node in the museum environment test.

In terms of efiiciency, a comparison between EM-LOC and the three competing approaches are reported in the right three subplots of Fig. 7.9. As is evident from the results of A2, the simple means of using the side-channel information as a prior estimate for selecting nodes in which to localise is not very effective. At low values of k localisation is attempted in many nodes, leading to low efficiency and slow performance, while at high k, nodes in which localisation would have been accurate are missed. The reason for this can easily be seen from 7.5 where it is evident that the WiFi feature similarity does not perfectly mirror the true metric similarity. From the figure it is clear that EM-LOC successfully localises significantly more incoming images than the competing approaches at the same k values. Another noticeable trend is that even at $k = 1$, EM-LOC is still able to localise most images, meaning that the localisation can be performed extremely efficiently (every incoming frame only has to be matched against the 3D features contained by a single node). The localisation efficiency of EM-LOC is extremely high, and thus the efficency is significantly higher for EM-LOC even compared to the A3 where a trained SVM is used to select candidate

nodes.

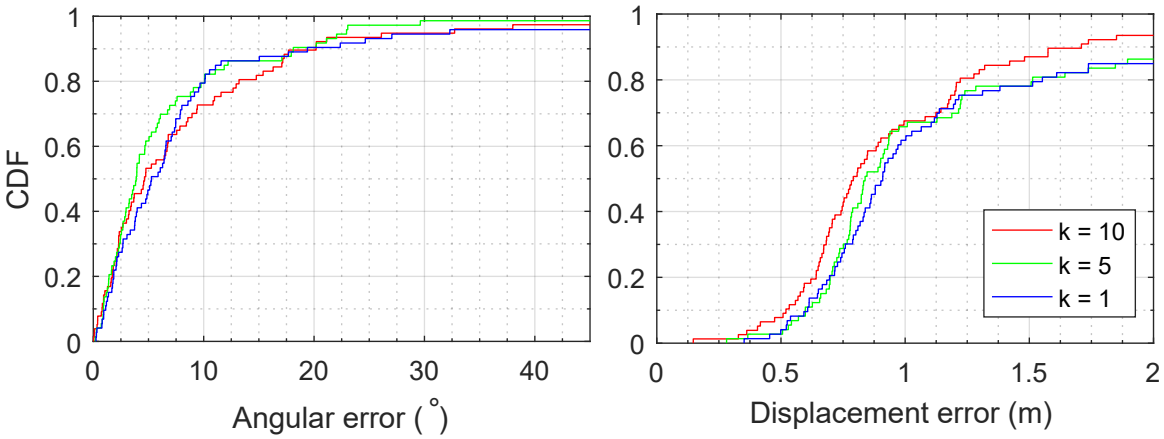


Figure 7.8: Cdf plot of the orientaation and translation error of EM-LOC from various values of k.

In Fig. 7.6 we report a comparison and breakdown of the on-line localisation processing times for EM-LOC and the three competing methods. Due to the efficiency EM-LOC significantly cuts down the processing time spent on 2D- 3D feature matching compared to A1 which uses all nodes in the matching process.

Table 7.1: Efficiency of the 6-DoF localisation for the 3 methods

Method	k = 10	k = 5	k = 1
EM-LOC	0.4783	0.4534	0.4534
A1	0.1043	0.0519	0.0187
A3	0.1317	0.1257	0.1078

In summary, by localising only against a select subset of nodes we are able to perform 6-DoF pose updates more frequently than exhaustive matching and also avoid occasional visual false positives, suppressing outlier pose estimates and resulting in a higher localization accuracy. Although the side-channel information is an ideal means of selecting candidate nodes for localisation, the noisy nature of this data means that existing methods such as using independent features (as in A2) or even an SVM that models correlations between features (as in A3) is not able to provide sufficiently accurate candidate nodes. By making use of a stream of side-channel information

along with rough odometry estimates the EM-LOC system can significantly boost the accuracy of the node proposals.

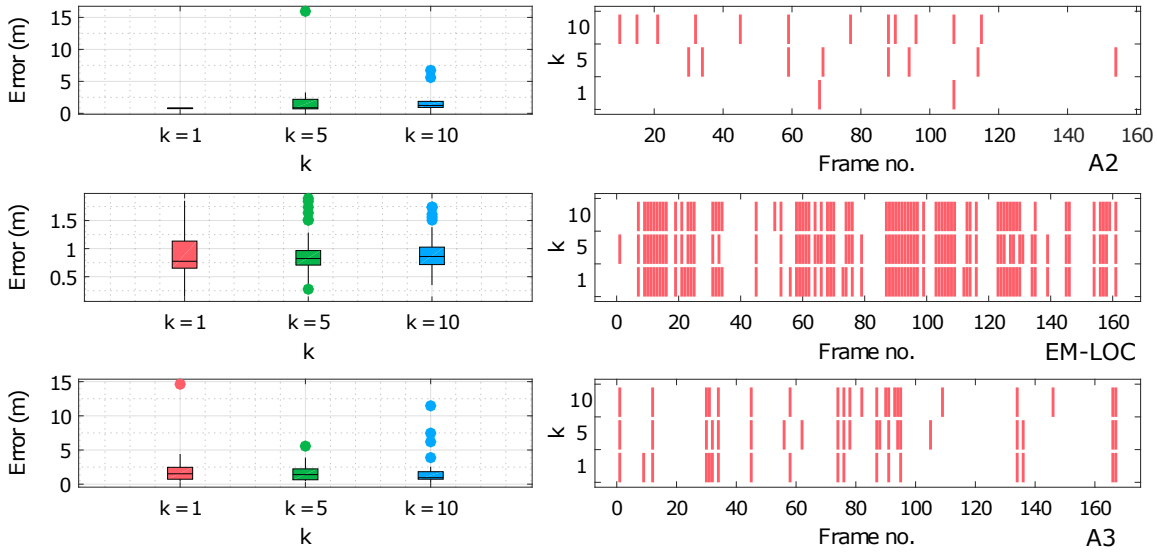


Figure 7.9: Evaluation of the localisation accuracy and the localisation efficiency. The right subplots show the successful localisations for each of the input test images.

7.7 Conclusion

In this chapter we have presented a method for performing efficient localization in large scale point clouds. Our method is based on exploiting auxiliary sensing information which is able to provide cues as to where a successful visual localization will occur. The method significantly reduces the number of visual pose computations that need to be carried out while keeping the accuracy of the estimation on par with the exhaustive pose evaluation at each timestep.

Chapter 8

Conclusions

In this thesis we have focussed on the problem of performing vision-based egocentric mapping and localization. The main challenges for visual localization and mapping systems on mobile platforms are *robustness, accuracy and efficiency.* As we have described, existing vision-based localization approaches readily fall short of these goals when operated in adverse, changing environments and making robust in these conditions remains an important and open research area. In this thesis, we have proposed *learning* as a possible means of achieving these goals. To this extent, we have made three major contributions; we proposed novel neural network architectures for performing local pose tracking (i.e. visual odometry), and re-localization that are competitive to traditional sparse feature based approaches; we have designed a mapping system which integrates the predictions of these deep networks with a traditional online optimization and loop-closure approach to reduce drift, further increase the localization accuracy and robustly create large scale maps. Finally, we have addressed the problem of efficiently and accurately localizing in these large-scale point-clouds. We have demonstrated the proposed approaches on real-world datasets. We conclude that learning-based approaches for localizaiton and mapping is an incredibly promising research direction which will help in bringing accurate

visual localization to a wide class of mobile platforms - from consumer devices to robotic systems.

8.1 Future work

The work presented in this thesis opens up a number of future research problems. These include:

1. **Self-supervised training of odometry network.** In this thesis we introduced a supervised and semi-supervised approach to training for out visual-inertial odometry network. Future work could consider fully self-supervised training which could even be used as a pre-training method if labelled data are available. One way to accomplish this would be to introduce a depth prediction and warping component in the network as done on a frame-to-frame basis in [170], allowing it to be trained on photometric error alone.

2. **End-to-end training of SLAM with online optimization of residuals.** Our integrated SLAM system with the learning-based priors considers the training of the networks for the prior predictions independently of the online optimization of the visual and inertial residuals. This approach was taken as the components of the standard SLAM system, such as the sparse feature matching, are non-differntiable. One avenue for future work may include integrating a online optimization of the reprojection residuals in a manner similar to the unsupervised training procedure suggested previously. However, more research is required to investigate the utility of such an approach as training through the optimization process may weaken the learning signal for the prior prediction networks.

3. **Dynamic objects and occlusions.** This thesis focussed on tracking and localizing the device itself from an ego-centric perspective. Although the presented

approach is fairly robust to moving obstacles in the image, it may be desirable in many cases to track such objects independently of the platform's motion. To some extent, our learning-based models for relocalization and ego-motion estimation do not make the strict assumption that all elements in the scene are static (eg. they could learn to reject moving objects) which is already better than approaches such as [15, 160] which assume a static map. However, future work could investigate learning-based approaches to explicitly identifying and tracking moving obstacles. Existing approaches which do this include Bibby and Reid [9] proposed Simultaneous Localisation and Mapping in Dynamic Environments (SLAMIDE) with Reversible Data Association which combines the least-squares optimization of a Graph-SLAM based method with expectation maximization to incorporate static and dynamic elements. A sliding window approach allows data association to be delayed, and possibly reversed, before a solid commitment to landmarks is made. There have also been various theoretical investigations into Multi-body Structure from Motion [158] and some research addressing the practical implementation of the MSfM problem [158]. However, MSfM techniques have not yet matured as processing can take as long as 1-minute per frame. Future work could include extending our integrating system into an non-rigid reconstruction system with robust data-driven priors.

Bibliography

[1] Motilal Agrawal and Kurt Konolige. Rough terrain visual odometry. In *Proceedings of the International* , 2007.

[2] Galen Andrew, Raman Arora, Jeff A Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *ICML (3)*, pages 1247–1255, 2013.

[3] Relja Arandjelovic and Andrew Zisserman. All about vlad. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013.

[4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[5] Yoshua Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[6] Yoshua Bengio, Yann LeCun, et al. Scaling learning algorithms towards ai. *Large-scale kernel machines*, 34(5), 2007.

[7] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: A cpu and gpu math compiler in python. In *Proc. 9th Python in Science Conf*, pages 1–7, 2010.

[8] Simone Bianco, Marco Buzzelli, Davide Mazzini, and Raimondo Schettini. Logo recognition using cnn features. In *International Conference on Image Analysis and Processing*, pages 438–448. Springer, 2015.

[9] C Bibby and I Reid. Simultaneous localisation and mapping in dynamic environments (slamide) with reversible data association, 2007.

[10] Christopher M Bishop. Mixture density networks. 1994.

[11] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. Robust visual inertial odometry using a direct ekf-based approach. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 298–304. IEEE, 2015.

[12] Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20, pages 161–168. NIPS Foundation (http://books.nips.cc), 2008.

[13] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, et al. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3364–3372, 2016.

[14] G. Bradski. opencv. *Dr. Dobb's Journal of Software Tools*, 2000.

[15] Wolfram Burgard, Armin B. Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. Experiences with an interactive museum tour-guide robot, 1999.

[16] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro

aerial vehicle datasets. *The International Journal of Robotics Research*, page 0278364915620033, 2016.

[17] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J.J. Leonard. Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age. *IEEE Transactions on Robotics*, 32(6):13091332, 2016.

[18] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[19] Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *arXiv preprint arXiv:1605.02305*, 2016.

[20] Nicholas Carlevaris-Bianco and Ryan M Eustice. Learning visual feature descriptors for dynamic lighting conditions. In *Proceedings of the {IEEE}/{RSJ} International Conference on Intelligent Robots and Systems*, pages 2769–2776, 2014.

[21] R. O. Castle, G. Klein, and D. W. Murray. Combining monoslam with object recognition for scene augmentation using a wearable camera. *Image and Vision Computing*, 28:1548–1556, 2010.

[22] Robert O Castle, Georg Klein, and David W Murray. Combining monoslam with object recognition for scene augmentation using a wearable camera. *Image and Vision Computing*, 28(11):1548–1556, 2010.

[23] W. Churchill and P. Newman. Experience-based navigation for long-term localisation. *The International Journal of Robotics Research*, 32(14):1645–1661, 9 2013.

[24] Ronald Clark, Sen Wang, Hongkai Wen, Andrew Markham, and Niki Trigoni. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[25] Lee E Clement, Valentin Peretroukhin, Jacob Lambert, and Jonathan Kelly. The battle for filter supremacy: A comparative study of the multi-state constraint kalman filter and the sliding window filter.

[26] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[27] Alejo Concha and Javier Civera. Dpptam: Dense piecewise planar tracking and mapping from a monocular sequence. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 5686–5693. IEEE, 2015.

[28] Alejo Concha and Javier Civera. Rgbdtam: A cost-effective and accurate rgb-d tracking and mapping system. *arXiv preprint arXiv:1703.00754*, 2017.

[29] Peter Corke and Pieter Abbeel. Deep learning: Are the sceptics right? *RSS Workshops 2016*, 2016.

[30] Gabriele Costante, Michele Mancini, Paolo Valigi, and Thomas A Ciarfuglia. Exploring representation learning with cnns for frame-to-frame ego-motion estimation. *IEEE Robotics and Automation Letters*, 1(1):18–25, 2016.

[31] M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance, 2008.

[32] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[33] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007.

[34] Frank Dellaert. Factor graphs and gtsam: A hands-on introduction. 2012.

[35] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016.

[36] Changxing Ding and Dacheng Tao. A comprehensive survey on pose-invariant face recognition. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(3):37, 2016.

[37] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, 2014.

[38] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016.

[39] Arnaud Doucet, Nando De Freitas, Kevin Murphy, and Stuart Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 176–183. Morgan Kaufmann Publishers Inc., 2000.

[40] M. Douze, H. Jégou, H. Singh, L. Amsaleg, and C. Schmid. Evaluation of gist descriptors for web-scale image search. In *ACM International Conference on Image and Video Retrieval*, pages 0–7, 2009.

[41] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[42] Felix Endres, Jürgen Hess, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. 3-d mapping with an rgb-d camera. *IEEE Transactions on Robotics*, 30(1):177–187, 2014.

[43] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. In *arXiv:1607.02565*, July 2016.

[44] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. September 2014.

[45] Jakob Engel, Thomas Sch, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *ECCV*, pages 1–16, 2014.

[46] Jakob Engel, Jurgen Sturm, and Daniel Cremers. Semi-dense visual odometry for a monocular camera. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1456, 2013.

[47] Ramsey Faragher and Robert Harle. An analysis of the accuracy of bluetooth low energy for indoor positioning applications. In *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014), Tampa, FL, USA*, volume 812, page 2, 2014.

[48] B Ferris, D Haehnel, and Dieter Fox. Gaussian processes for signal strength-based location estimation. *In Proc. of Robotics Science and Systems*, 442:303–310, 2006.

[49] Brian Ferris, Dieter Fox, and Neil D Lawrence. Wifi-slam using gaussian process latent variable models. In *IJCAI*, volume 7, pages 2480–2485, 2007.

[50] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *arXiv preprint arXiv:1504.06852*, 2015.

[51] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In *Robotics: Science and Systems XI*, number EPFL-CONF-214687, 2015.

[52] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, 5 2014.

[53] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22. IEEE, 2014.

[54] Martin Frassl, Michael Angermann, Michael Lichtenstern, Patrick Robertson, Brian J Julian, and Marek Doniec. Magnetic maps of indoor environments for precise localization of legged and non-legged locomotion. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 913–920. IEEE, 2013.

[55] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, 43(1):55–81, 2015.

[56] Paul Furgale and Timothy D. Barfoot. Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics*, 27:534–560, 2010.

[57] T Gadeke, Johannes Schmid, Wilhelm Stork, and KD Muller-Glaser. Pedestrian dead reckoning for person localization in a wireless sensor network. In *Proc. 2011 International conference on indoor positioning and indoor navigation*, pages 1–4, 2011.

[58] Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015.

[59] D Gálvez-López and JD Tardós. Bags of binary words for fast place recognition in image sequences. *Robotics, IEEE Transactions on*, pages 1–9, 2012.

[60] Ravi Garg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016.

[61] Steffen Gauglitz, Tobias Höllerer, and Matthew Turk. Evaluation of interest point detectors and feature descriptors for visual tracking. *International journal of computer vision*, 94(3):335, 2011.

[62] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, page 0278364913491297, 2013.

[63] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.

[64] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 963–968, 2011.

[65] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. *arXiv preprint arXiv:1609.03677*, 2016.

[66] Thomas Goldstein, Paul Hand, Choongbum Lee, Vladislav Voroninski, and Stefano Soatto. Shapefit and shapekick for robust, scalable structure from motion. In *European Conference on Computer Vision*, pages 289–304. Springer, 2016.

[67] Jianjun Gui, Dongbing Gu, Sen Wang, and Huosheng Hu. A review of visual inertial odometry from filtering and optimisation perspectives. *Advanced Robotics*, 29(20):1289–1301, 2015.

[68] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and P-J Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on signal processing*, 50(2):425–437, 2002.

[69] Simon S Haykin, Simon S Haykin, and Simon S Haykin. *Kalman filtering and neural networks*. Wiley Online Library, 2001.

[70] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[71] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[72] Fu Jie Huang, Y-Lan Boureau, Yann LeCun, et al. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007.

[73] Guoquan Huang, Michael Kaess, and John J Leonard. Towards consistent visual-inertial navigation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4926–4933. IEEE, 2014.

[74] Arnold Irschara, Christopher Zach, Jan Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, pages 2599–2606, 2009.

[75] Satoshi Ito, Felix Endres, Markus Kuderer, Gian Diego Tipaldi, Cyrill Stachniss, and Wolfram Burgard. W-rgb-d: floor-plan-based indoor global localization using a depth camera and wifi. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 417–422. IEEE, 2014.

[76] Viren Jain, Joseph F Murray, Fabian Roth, Srinivas Turaga, Valentin Zhigulin, Kevin L Briggman, Moritz N Helmstaedter, Winfried Denk, and H Sebastian Seung. Supervised learning of image restoration with convolutional networks. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[77] Viren Jain and Sebastian Seung. Natural image denoising with convolutional networks. In *Advances in Neural Information Processing Systems*, pages 769–776, 2009.

[78] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

[79] Yunye Jin, Hong-Song Toh, Wee-Seng Soh, and Wai-Choong Wong. A robust dead-reckoning pedestrian tracking system with low cost sensors. In *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, pages 222–230. IEEE, 2011.

[80] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2016)*, 35(6), 2016.

[81] Anton Kasyanov, Francis Engelmann, Jörg Stückler, and Bastian Leibe. Keyframe-based visual-inertial online slam with relocalization. *arXiv preprint arXiv:1702.02175*, 2017.

[82] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2016.

[83] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2938–2946, 2015.

[84] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[85] Alexander Kleiner and Dali Sun. Decentralized slam for pedestrians without direct communication. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1461–1466. IEEE, 2007.

[86] Kishore Konda and Roland Memisevic. Learning visual odometry with a convolutional network. In *International Conference on Computer Vision Theory and Applications*, 2015.

[87] Till Kroeger and Luc Van Gool. Video registration to sfm models. In *European Conference on Computer Vision*, pages 1–16. Springer, 2014.

[88] Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 447–458. ACM, 2014.

[89] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[90] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155–166, 2009.

[91] Stefan Leutenegger, Paul Furgale, Vincent Rabaud, Margarita Chli, Kurt Konolige, and Roland Siegwart. Keyframe-based visual-inertial slam using nonlinear optimization. *Proceedings of Robotics: Science and Systems*, 2013.

[92] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[93] A Quattrini Li, A Coskun, SM Doherty, S Ghasemlou, AS Jagtap, M Modasshir, S Rahman, A Singh, M Xanthidis, JM OKane, et al. Experimental comparison of open source vision based state estimation algorithms. In *Int. Symp. on Experimental Robotics*, 2016.

[94] Bo Li, Chunhua Shen, Yuchao Dai, Anton van den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1119–1127, 2015.

[95] Chen Liang-Chieh, George Papandreou, Iasonas Kokkinos, kevin murphy, and Alan Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In *International Conference on Learning Representations*, San Diego, United States, May 2015.

[96] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[97] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[98] Ang Lu, Weiran Wang, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Deep multilingual correlation for improved word embeddings. In *Proceedings of NAACL*, 2015.

[99] Simon Lynen, Michael Bosse, Paul Furgale, and Roland Siegwart. Placeless place-recognition. In *3D Vision (3DV), 2014 2nd International Conference on*, volume 1, pages 303–310, 2014.

[100] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, to appear.

[101] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[102] Colin McManus, Winston Churchill, Ashley Napier, Ben Davis, and Paul Newman. Distraction suppression for vision-based pose estimation at city scales. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3762–3769, 2013.

[103] Sven Middelberg, Torsten Sattler, Ole Untzelmann, and Leif Kobbelt. Scalable 6-dof localization on mobile devices. In *European Conference on Computer Vision*, pages 268–283. Springer, 2014.

[104] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler. Online multi-target tracking using recurrent neural networks. *arXiv:1604.03635 [cs]*, April 2016. arXiv: 1604.03635.

[105] Michael Milford, Eleonora Vig, Walter Scheirer, and David Cox. Vision-based simultaneous localization and mapping in changing outdoor environments. *Journal of Field Robotics*, 31(5):780–802, 2014.

[106] Michael J. Milford and Gordon F. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1643–1649, 2012.

[107] Michael J Milford, Gordon F Wyeth, and David Prasser. Ratslam: a hippocampal model for simultaneous localization and mapping. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 403–408. IEEE, 2004.

[108] J. M M Montiel and Andrew J. Davison. A visual compass based on slam. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2006, pages 1917–1922, 2006.

[109] Anastasios I. Mourikis and Stergios I. Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3565–3572, 2007.

[110] Anastasios I Mourikis and Stergios I Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3565–3572. IEEE, 2007.

[111] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, page 15, 2015.

[112] Raul Mur-Artal, JMM Montiel, and Juan D Tardós. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[113] Raúl Mur-Artal and Juan D Tardós. Orb-slam: Tracking and mapping recognizable. In *Workshop on Multi VIew Geometry in Robotics (MVIGRO) - RSS 2014*, 2014.

[114] Ashley Napier and Paul Newman. Generation and exploitation of local orthographic imagery for road vehicle localisation. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 590–596, 2012.

[115] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[116] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011.

[117] Paul Newson and John Krumm. Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 336–343. ACM, 2009.

[118] CT Ng, MS Barketau, TC Edwin Cheng, and Mikhail Y Kovalyov. product partition and related problems of scheduling and systems reliability: Computational complexity and approximation. *European Journal of Operational Research*, 207(2):601–604, 2010.

[119] David Nistér and Henrik Stewénius. Scalable recognition with a vocabulary tree. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2161–2168, 2006.

[120] Gabriel Nützi, Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Fusion of imu and vision for absolute scale estimation in monocular slam. In *Journal of Intelligent and Robotic Systems: Theory and Applications*, volume 61, pages 287–299, 2011.

[121] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42:145–175, 2001.

[122] Geoffrey Pascoe, Will Maddern, and Paul Newman. Robust direct visual localisation using normalised information distance. 2015.

[123] Florent Perronnin, Yan Liu, Jorge Sanchez, and Herve Poirier. Large-scale image retrieval with compressed fisher vectors. In *2010 IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition, pages 3384–3391, 2010.

[124] Sudeep Pillai and John Leonard. Monocular slam supported object recognition. *arXiv preprint arXiv:1506.01732*, 2015.

[125] Jason R Rambach, Aditya Tewari, Alain Pagani, and Didier Stricker. Learning to fuse: A deep learning approach to visual-inertial camera pose estimation. In *Mixed and Augmented Reality (ISMAR), 2016 IEEE International Symposium on*, pages 71–76. IEEE, 2016.

[126] Ananth Ranganathan, Shohei Matsumoto, and David Ilstrup. Towards illumination invariance for visual localization. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3791–3798, 2013.

[127] Valérie Renaudin, Melania Susi, and Gérard Lachapelle. Step length estimation using handheld inertial sensors. *Sensors*, 12(7):8507–8525, 2012.

[128] Jose Rivera-Rubio, Ioannis Alexiou, Anil Bharath, Riccardo Secoli, Luke Dickens, Emil C Lupu, and UK South Kensington Campus. Associating locations from wearable cameras. In *Proceedings of the British Machine Vision Conference. BMVA Press*, 2014.

[129] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):105–119, 2010.

[130] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

[131] Zafer Sahinoglu, Sinan Gezici, and Ismail Guvenc. Ultra-wideband positioning systems. *Cambridge, New York*, 2008.

[132] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H J Kelly, and Andrew J. Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1352–1359, 2013.

[133] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1352–1359, 2013.

[134] Fendy Santoso, Matthew Garratt, Mark Pickering, and Md Asikuzzaman. 3d-mapping for visualisation of rigid structures: A review and comparative study. *IEEE Sensors*, 2015.

[135] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *2011 International Conference on Computer Vision*, pages 667–674. IEEE, 2011.

[136] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Improving image-based localization by active correspondence search. In *European Conference on Computer Vision*, pages 752–765. Springer, 2012.

[137] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, volume 6, page 7, 2012.

[138] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[139] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[140] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014.

[141] MA Shelley. *Monocular visual inertial odometry on a mobile device*. PhD thesis, Masters thesis, Technischen Universitat Munchen, 2014.

[142] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013.

[143] Yuanchao Shu, Cheng Bo, Guobin Shen, Chunshui Zhao, Liqun Li, and Feng Zhao. Magicol: indoor localization using pervasive magnetic field and opportunistic wifi sensing. *Selected Areas in Communications, IEEE Journal on*, 33(7):1443–1457, 2015.

[144] Ivan Sikirić, Karla Brkić, and Siniša Šegvić. Classifying traffic scenes using the gist image descriptor. *arXiv preprint arXiv:1310.0316*, 2013.

[145] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[146] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. *Proceedings Ninth IEEE International Conference on Computer Vision*, 2003.

[147] Shiyu Song, Manmohan Chandraker, and Clark C Guest. High accuracy monocular sfm and scale correction for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence*, 38(4):730–743, 2016.

[148] Henry Stark and Yongyi Yang. Vector space projections. *John Wiley&Sons, New York*, 1998.

[149] Alexander D. Stewart and Paul Newman. Laps - localisation using appearance of prior structure: 6-dof monocular camera localisation using prior pointclouds. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2625–2632, 2012.

[150] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 573–580. IEEE, 2012.

[151] Niko Sünderhauf and Peter Protzel. Switchable constraints for robust pose graph slam. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1879–1884. IEEE, 2012.

[152] Niko Sunderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems XII*, 2015.

[153] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[154] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[155] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. *arXiv preprint arXiv:1704.03489*, 2017.

[156] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.

[157] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692. ACM, 2015.

[158] René Vidal and Richard Hartley. Three-view multibody structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:214–227, 2008.

[159] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017.

[160] DF Wolf and GS Sukhatme. Semantic mapping using mobile robots. *Robotics, IEEE Transactions on*, 24(2):245–258, 2008.

[161] Changchang Wu. Towards linear-time incremental structure from motion. In *3D Vision-3DV 2013, 2013 International Conference on*, pages 127–134. IEEE, 2013.

[162] Katsu Yamane. Systematic derivation of simplified dynamics for humanoid robots. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 28–35. IEEE, 2012.

[163] Fei Yan and Krystian Mikolajczyk. Deep correlation for matching images and text. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3441–3450, 2015.

[164] Jun Yang, Yu-Gang Jiang, Alexander G Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 197–206. ACM, 2007.

[165] Georges Younes, Daniel Asmar, Elie Shammas, and John Zelek. Keyframe-based monocular slam: design, survey, and future directions. *Robotics and Autonomous Systems*, 98:67–88, 2017.

[166] Wojciech Zaremba. An empirical exploration of recurrent network architectures. 2015.

[167] Wei Zhang and Jana Kosecka. Image based localization in urban environments. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 33–40. IEEE, 2006.

[168] Yuanqing Zheng, Guobin Shen, Liqun Li, Chunshui Zhao, Mo Li, and Feng Zhao. Travi-navi : Self-deployable indoor navigation system. In *Proceedings Mobicom 2014*, number 2. ACM, 2014.

[169] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.

[170] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. *arXiv preprint arXiv:1704.07813*, 2017.