Dimensionality Reduction with Subpixel Refinement for SLAM

Dinesh Gamage

Department of Electrical and Computer Systems Engineering

Monash University

PhD Thesis

2015

Supervisor: Prof. Tom Drummond

Declaration

This dissertation is submitted to Monash University Australia in partial fulfilment of the degree of Doctor of Philosophy. It is an account of work undertaken at the Department of Electrical and Computer Systems Engineering between March 2011 and March 2015 under the supervision of Professor T.W. Drummond. It is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.

.....

Dinesh Gamage

Acknowledgements

I thank my supervisor Professor Tom Drummond for his guidance and highly intellectual and motivating discussions.

I also would like to thank the Monash University for funding my work.

Finally I would like to thank my parents, sister, brother in-law and colleagues for the enormous emotional support they provided.

Publications

- D.Gamage, T.Drummond. Corner Matching Refinement for Monocular Pose Estimation. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2012.
- D.Gamage, T.Drummond. Reduced Dimensionality Extended Kalman Filter for SLAM. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2013.
- K.Ok, D.Gamage, T.Drummond, F.Dellaert, N.Roy Monocular Image Space Tracking on a Computationally Limited MAV. In *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2015.
- D.Gamage, T.Drummond. Reduced Dimensionality Extended Kalman Filter for SLAM in a Relative Formulation. In *Proceedings of the International Conference on Intelligent Robots and Systems*. IEEE/RSJ, 2015.

Abstract

A simultaneous localisation and mapping (SLAM) system continuously explores the environment to causally estimate the ego-motion of a robot and map the environment. Visual SLAM using a single video camera is particularly challenging. Although visual SLAM allows incorporating thousands of features into the system to improve the accuracy, this gain comes with a computational overhead. This thesis advances the state of the art in visual SLAM in terms of efficiency, accuracy and robustness.

First, a sub-pixel refinement algorithm is presented to permit efficient pose estimation in monocular SLAM. The algorithm extends spatial domain Gauss-Newton parameter estimation into the frequency domain. Then corresponding features are sub-pixel refined by estimating the affine parameters between the two surrounding patches. Here, the correct frequency range is selected by identifying a direct relationship between the Gabor phase response and the frequency response of a Gaussian multiplied image patch. Further it is shown how parameter estimation can be made more accurate by operating in the frequency domain, which naturally gives rise to a multi-resolution optimisation framework.

Next, a novel method is proposed to handle the dimensionality of the SLAM problem which permits the handling of a large number of parameters. The proposed method dramatically reduces the computational complexity of the Kalman-filters by reducing the dimensionality as information is acquired. The validity of the method is proved

by applying it to monocular SLAM, where there are a large number of dimensions in the filter that are not subject to process noise (the landmark locations). This has the effect of reducing the cost of running a filter or allowing a single filter to process a much larger set of landmarks.

Then, the dimensionality reduction is extended into a relative formulation, which is extensible into a large-scale system. The formulation uses the higher degree of linearity available with the relative formulation to build a Kalman-filter based reduced SLAM system. An un-delayed method for adding features to the filter is also introduced. Then the effect of the number of features used in the system on the final estimation uncertainty is analyzed, and it is shown that the actual number of dimensions that has to be optimised is far less than the number of original dimensions in the problem.

Finally, we introduce a novel method to retrieve the pose estimation Jacobian on limited platforms through an efficient partitioning of the matrix, which removes the Jacobian computation overhead. Instead of recalculating the Jacobian every time, we show how it can be precalculated and saved for later retrieval.

Contents

List of Figures

Chapter 1

Introduction

In this Chapter first we give a brief introduction SLAM. A more thorough analysis can be found in Chapter 2. Then we introduce the main contributions of this thesis.

1.1 Simultaneous Localisation and Mapping

In robotics, SLAM is the computational problem of constructing a map of an unknown environment while simultaneously tracking the trajectory of the robot in that environment. As humans, we can estimate the location of ourselves within a given environment very easily through relative measurements of the objects, which we perceive through our eyes.

A moving robot acquires information related to its environment using its sensors. The environment is usually described using physical features. Such features can be points, edges or any other form of physical cues that can be easily ex-

tracted from the environment and then tracked to obtain a measurement of the relative transformation the robot experiences due to its own movement. Tracking is necessary to measure the movement of the robot relative to those features. Due to sensor noise it is impossible to compute the actual state of the system directly. As a result, the robot has to maintain a belief about its state. The state is the set of parameters the robot maintains to describe both its environment composed of landmarks and the relative poses of the its trajectory. If the environment is known, determining the pose of the robot, or given the robots trajectory, mapping its environment is quite straightforward. In an unknown environment, simultaneously localising the pose while mapping the environment is rather complicated. There will always be an uncertainty associated with the robots belief due to sensor noise. The accuracy of the estimation depends on the nature of the underlying algorithm and the sensors used.

The camera as a sensor has received much attention in robotics as a low-cost and feature-rich alternative for SLAM. A single camera is a bearing-only sensor, which is small, reliable and cheap. However, bearing-only SLAM is hindered by the feature initialisation problem, where the depth of a newly mapped landmark cannot be determined from a single measurement, demanding multiple bearing-only measurements in combination. Combined information from multiple measurements may be ill-conditioned, making the estimation much more complex compared to estimating those with bearing and range sensors. If two or more cameras are used together to obtain measurements, the range information can be directly computed.

1.1.1 Monocular SLAM

Monocular SLAM is the particular problem where the only sensor used is a single camera. As a single camera is a bearing-only sensor, the feature depths and the robot poses can only be recovered up to a scale as there is no reference available at the initialisation stage to make relative measurements. The scale of the system

is usually governed by the scale that the system assumes at initialisation.

Visual SLAM relies on image feature matching. The corresponding features are matched between images to compute the pose. In recent dense systems, those correspondences can even be all pixels of the images. After matching a sufficient number of correspondences, the relative camera pose between such frames can be computed using a minimal set of five points in closed form or incrementally. The estimated pose then can be used to estimate the depth of matched features in the monocular case, or to refine the measured depths if it is available.

After feature initialisation, a robot can continuously estimate its ego-motion using inter-frame feature matches. Already initialised features which appear again on a new frame are used to estimate the relative pose of that frame. If the number of initialised features that appear on the new frame drops below a given threshold, a new set of features is initialised where the depths can be estimated using already estimated relative pose (with respect to those initialised landmark correspondences) of the new frame. The optimisation is done simultaneously with estimating the camera pose while refining the landmark estimates.

In visual SLAM, the features on a frame are described using the colour or the texture of the surrounding area of a given feature and then matched by searching for similar colour or texture patterns in another frame. The information extracted in this manner to describe a given feature is called a descriptor. Changing the viewing location, which changes the perspective projection, can lead to mismatching of such features, as the descriptor also changes due to the projection. There are different methods to make the descriptors more robust to perspective changes. However sophisticated the method is, it will consume more processing time, hence the type of the descriptor has to be chosen depending on the application. Even if the descriptor is reasonably invariant to perspective projections, there can be mismatches which would corrupt the final estimation. If the number of correct matches of such a system is much greater than the number of mismatches, random sample consensus (RANSAC) is used to select a reasonable estimation. RANSAC selects a camera pose which agrees with the majority of the matches from a set

of pose hypotheses estimated from different subsets of feature matches selected randomly.

The inter-frame pose computed in this manner cannot maintain the consistency of a SLAM system when the map grows, as such estimates do not account for the uncertainties associated with the measurements. The sensor noise makes the landmarks and the camera poses inter-dependent on each other. With association of uncertainties, the solution of such a system can be broadly categorised into two paradigms: filtering and bundle adjustment.

In filtering, the uncertainty relationships are represented explicitly using a covariance matrix or an information matrix. This allows updating of the system through sequential measurements. In bundle adjustment, the measurements starting from the beginning have to be used as a batch to solve a linear system of equations. Although it is not explicitly maintained, the covariance structure of the filtering approach implicitly becomes embedded into bundle adjustment through its constituent equations, making both methods theoretically equivalent.

1.2 Contributions

1.2.1 Corner Matching Refinement for Monocular Pose Estimation

Because of the inter-dependent nature of parameters in SLAM, reducing the measurement noise affects the accuracy of the whole system. To improve the accuracy of the system, descriptor-based feature matching which works at pixel level may not be adequate and sub-pixel level information may be needed. A number of applications already use something better than pixel-sampled information. Al-

though sub-pixel methods have been used extensively for stereo matching, most of these techniques are based on the assumption that the 2-D image motion, resulting from 3-D camera motion can be described using a simple translation model.

As a solution, in Chapter 3 we present a novel method for refining the coordinates of correspondences directly. Given some coordinates in the first image, our goal is to maximise the accuracy of the estimate of the coordinates in the second image corresponding to the same real world point without being too concerned about which real world point is being matched. We parametrise the image signal around a selected feature using the six-parameter affine model with an additional parameter to compensate for energy changes of the signal. In pose estimation, since the illumination between two consecutive frames will not change significantly, the effect of the seventh parameter is trivial for our application. By optimising in the frequency domain, it is possible to achieve improved results and a faster convergence rate. The fast convergence is a result of the multi-resolution nature that naturally arises with such an approach, as explained later.

1.2.2 Reduced Dimensionality Extended Kalman Filter for SLAM

Filtering cannot be easily extended to large scales as the complexity of the filter grows rapidly, making it extremely difficult to maintain a large covariance matrix explicitly. To handle this complexity to some extent, filter-based SLAM systems maintain only the current pose estimate with all the landmarks of interest as the state. This representation marginalises out previous camera poses, connecting all state elements with each other, making the covariance matrix dense, leading to a fully connected graph. Because of this, the filter complexity grows at least quadratically with new observations, quickly making the problem intractable.

The quadratic growth can be handled by dividing the whole problem into small tractable sup-maps while separately maintaining global consistency. Developments like parallel tracking and mapping (PTAM) and dense tracking and mapping (DTAM) take this further, by noticing the possibility of working with an active set of landmarks by separating tracking from mapping. In these systems, the tracker works with a local set of landmarks while the map maintains the global consistency through bundle adjustment. Full bundle adjustment in PTAM adjusts the poses using the measurements of all key-frames. It exploits the sparseness inherent in the structure from the motion problem to reduce the computational complexity. Without maintaining all landmark descriptors in this manner, one could even use a more efficient sparse matrix system as the back end to build a globally consistent map. As efficient as sparse matrix methods are, they still have limitations and are not used to process all frames of a video, as this would generate much denser graphs with high connectivity which would overwhelm these approaches. Hence they are restricted to using sparsely sampled key frames.

To overcome the complexity of filter-based SLAM, Chapter 4 explains a method to reduce the dimensions which are not subject to process noise of the extended Kalman filter (EKF) by identifying dominant modes of the filter. The method can be used in general to reduce the dimensionality of the EKF irrespective of its application without being limited to SLAM. In Chapter 4 we describe a method to reduce the dimensionality of each node of a graph-based SLAM system, as the graph optimisation problem we are considering is nearly linear with the number of nodes and as it combines the information from multiple frames to give richer key frames. The most complex operation in such a system is the Kalman filtering within each node, which imposes an upper limit on the maximum number of landmarks that each node can handle. Our approach can reduce the dimensionality of Kalman filters used in each node, increasing the number of features a node can handle leading to a more accurate estimation. As we suggest in the discussion, such a dimensionality-reduced graph can even be used in a multi-camera set-up more efficiently than a key-frame based sparse matrix method.

Here we consider the dimensionality reduction of the conventional extended Kalman filter for SLAM, where all past camera poses are marginalised out from the system, making the only camera pose maintained in the system change dynamically. Therefore the only variables that are not subject to process noise will be landmarks. We show how to reduce the dimensionality of such a system by treating the parameters of the camera separately from the landmarks.

1.2.3 Reduced Dimensionality Extended Kalman Filter for SLAM in a Relative Formulation

Marginalising previous camera poses from the EKF leads to the particular problem of linearization errors being permanently embedded into the system, as those previous poses which had been linearised at a particular point, which might not be correct, cannot be re-linearised to accommodate information acquired through new measurements. This makes the system deviate from the correct solution, making the assumed Gaussian uncertainty distribution invalid.

To avoid the non-linearity problem, particle-filtering has been used in the filtering paradigm to efficiently represent multi-model distributions. However the exponential growth of the number of particles with the dimensionality of the problem limits applicability of the particle-filter in real-time systems. Bundle adjustment, on the other hand, keeps the whole camera trajectory thus avoiding the non-linearity problem.

With the dimensionality reduction technique described in Chapter 4, system complexity can be maintained at a manageable level. The system described previously assumes marginalisation of previous camera poses, so the non-linearity problem will continue to exist. However, as we can reduce the dimensionality, it is possible to retain all camera poses without marginalising them. This enables re-linearization to accommodate new information. Chapter 5 presents a novel rel-

ative formulation of the EKF-SLAM in a reduced dimensional framework, where all previous camera poses are maintained in the system.

Although recent developments have shown that the amount of information fused improves the accuracy of the system, still this improvement has not been quantitatively analysed. We investigate further the effect of the amount of information on the uncertainty of the system.

The relative formulation represents each landmark relative to its initialised coordinate frame instead of using a globally-privileged coordinate system. This representation greatly relieves the system from its inherent non-linearity limitation. We also show that this method, which uses measurements from all camera frames, can yield more accurate results compared to systems like double window optimisation, which assumes some spatial or structural sparseness.

1.2.4 Monocular Image Space Tracking on a Computationally Limited MAV

Chapter 6 introduces an efficient-front end for monocular SLAM. This work has been completed as a collaborative project with the **Center for Robotics and Intelligent Machines (RIM) at Georgia Institute of Technology** and the **Computer Science and Artificial Intelligence Laboratory (CSAIL) at Massachusetts Institute of Technology**.

The front-end is used for camera-based navigation for computationally limited micro air vehicles (MAVs). Our approach is derived from the recent development of parallel tracking and mapping algorithms. Unlike previous results, we show how the tracking and mapping processes can operate using different representations. This reduces the computational cost of using the same model for both. Our primary contribution is to show how the cost of tracking the vehicle pose

can be substantially reduced by estimating the camera motion directly in the image space rather than in the world co-ordinate space. Here we estimate the camera motion efficiently through a novel Jacobian partitioning method enabling real time on-board feature tracking.

Chapter 2

Background

This chapter explains the advancements in SLAM starting from the basics. Here we describe important work that has been done during past few years in the fields relevant to the material presented in this thesis.

2.1 Simultaneous Localisation and Mapping

As mentioned in the introduction, the full-state of a SLAM system consists of the camera trajectory and the landmarks. The landmarks can be any thing that can be readily extracted from the environment to represent a particular part or a section of it. Therefore, the landmarks can be a set of $3-D$ points, lines or other objects which can be tracked continuously to obtain a relative estimation of the camera pose. Pose changes in $3-D$ can be represented as rigid body transformations in $SE3$, or if the scale matters as a similarity transformation in $Sim3$.

The effect of sensor noise makes it necessary to represent the state of the system probabilistically. In other words, at a given time what we have is only an understanding about the true state of the system to some level of certainty, which can be described only as a probabilistic value rather than an absolute state. When more measurements are made our understanding increases, in turn increasing the probability of that being correct.

2.1.1 Probabilistic Framework for SLAM

As one of the earlier probabilistic frameworks for SLAM, the seminal paper by Smith et al. [1988] describes a representation of spatial relationships of SLAM which makes their inherent uncertainties explicit. In this framework, all spatial relationships are tied together in a representation called a stochastic map. It contains estimates of spatial relationships, their uncertainties and the interdependencies of the state. The proposed state contains the current robot location in $2-D$ and the feature locations extracted from the environment which in contrast to maintaining the full trajectory retains only the current pose as it has been used widely in the earlier systems as explained in following paragraphs. In this representation, the camera pose and the landmark estimates are formed into a vector called the state and their uncertainties are represented using an explicit covariance matrix. The diagonal values of the covariance matrix represent the variance of a variable with itself, while the off-diagonals represent the variances with each other. Newly initialised landmarks are appended to the state using a set of new parameters with a large covariance block appended to the covariance matrix. The initial covariance is set to a large value as there is a larger uncertainty associated with the belief of the new variable. In particular, the main contribution of this work is showing that the first and the second moments of the actual distribution are sufficient for reasonably modelling the relationships and quantities of the map.

Computing the actual values of the spatial variables requires knowledge of the

complete probability density function, which will not generally be available. This is caused by the non-linear relationships that exist between the system variables. The usual approach is to approximate the non-linear function through a truncated Taylor expansion, as as shown by Smith et al. [1988] and Durrant-Whyte [1988]. The Jacobians are always understood to be evaluated at the estimated mean of the given variables. The covariance also has to be transformed using the Jacobian of the function at the mean.

As the camera is moving, Smith et al. [1988] use a process model which describes how components of the system's state vector change (as a function of time in a continuous system, or by discrete transitions). The process model extrapolates the current state estimate to obtain a prediction of the next state by applying the dynamics of the system. When spatial information is available, that information is fused into the system to update this predicted state. The new information is added through a measurement model. With respect to new measurements, the estimations can be represented as a probability distribution given the measurements. Estimating the new mean in this manner naturally leads to a Kalman filtering (Kalman [1960]) framework. The information transfer of a Gaussian distribution remains Gaussian for a linear system, so that the result is optimal. Approximating non-linear systems through a truncated Taylor series expansion leads to a sub-optimal solution to the problem, as shown by Maybeck [1982].

The work by Leonard and Durrant-Whyte [1992] also uses an extended Kalman filter, where a sparse set of features extracted using its sensors is used to represent the environment. Information is fused into the system using a measurement model through a linearised projection function. After each measurement, the next state is predicted using a dynamic model. Although this system is quite similar to the system by Smith et al. [1988], the key insight is the use of only a subset of landmarks which are visible at a time. This makes the Jacobian of the observation model sparse, making the computations less intense.

The work by Davison et al. [2007] uses a Kalman filter for SLAM in a purely

vision set-up using a single camera. The key contribution of this work is to show that it is possible to achieve real-time localisation and mapping with a single freely moving camera as the only data source. The method achieves the efficiency required for real-time operation by using an active guided approach to feature mapping and measurement, a general motion model for smooth $3-D$ camera movement to capture the dynamical priori information inherent in a continuous video stream, and a novel top-down solution to the problem of monocular feature initialisation. The feature matching is done by restricting the search area on the image by gating the uncertainty of each landmark onto the camera image through the measurement function. Marginalising the previous camera poses to maintain only the current pose significantly reduces the complexity. However the total size of the map representation is still in the order of $\mathcal{O}(N^2)$, where N is the number of features and the complete SLAM algorithm has a $\mathcal{O}(N^2)$ complexity. This means that the number of features which can be maintained with real- time processing is bounded to around 100.

In attempts to improve the scalability of filter-based SLAM, Eustice et al. [2005] showed that there is a special structure in the covariance matrix when it is viewed in its information form. A filter which uses an information matrix is called an information filter. By exploiting this special structure, they managed to improve the scalability of the problem, where the filter which arises is called the sparse extended information filter (SEIF). The key to scalability in the information form are the strong and weak constraints of the information matrix which naturally arises in SLAM. It has been shown that the time-projection creates the weak constraints. Furthermore, by bounding the number of non-zero off-diagonal elements those link the camera pose with landmarks, it was possible to enforce an exact sparsity in the information matrix. The method maintains two sets of landmarks as active features and passive features where the weak links created between the camera pose and those passive landmarks are pruned. By enforcing an upper bound on the number of active features, it is possible to control the resulting fill-ins of the information matrix. However, ignoring the dependence relationships in, the SEIF sparsification strategy leads to inconsistent covariance estimates. Rather than heuristically ignoring the weak links between the camera

and the inactive landmarks, Walter et al. [2007] gave a more probabilistic approach to the problem to improve the inconsistency of covariance estimates. They termed this framework the exactly sparse extended information filter (ESEIF). This work showed that there is an implied conditional independence between the inactive landmarks and the camera, given the active landmarks. Generally, as a consequence of this enforced sparsification of information filters, the SLAM posteriori significantly underestimates the uncertainty of the state estimate. It has been shown that this inconsistency is a natural consequence of imposing conditional independence between the robot pose and a subset of the map. The main contribution of these systems is in showing the inherent sparse structure of a SLAM system which is not directly evident from the conventional Kalman filter.

Despite the scalability problem, marginalising previous camera poses from the system leads to inconsistencies of the final solution, as shown by Bailey et al. [2006a] and many other authors. At each marginalisation step, the linearization error of the marginalised variable is backed into the system, making the final solution drift from the absolute, hence deviating the system from the assumed Gaussian probability distribution and making the filter invalid.

In order to model non-Gaussian distributions, by extending the insight of ESEIFs, Montemerlo et al. [2002] introduced a factored solution to the simultaneous localization and mapping problem (FastSLAM), which showed that all individual landmark estimation problems are independent if one know the robots path and the correspondence variables. The system handles the deviation from a Gaussian distribution through a particle filtering approach, where each particle represents a guess of the robot's path. FastSLAM represents the conditional landmark estimates using Kalman filters, where the update is done through extended Kalman filters. In this case, the observation model remains Gaussian as a consequence of the use of sampling to approximate the distribution over the robots pose. Although this solves the non-linearity problem, with the number of particles the complexity of the system keeps growing, limiting the scalability of the system.

In contrast, the structure from motion (SFM) maintains all camera poses and

landmarks with their measurements to form a large system of equations. The main advantage of SFM is the possibility of re-linearization of the Jacobian, as all parameters are preserved in the system, yielding the ideal solution. However, in the original form, the system grows very fast and soon becomes intractable. With improved computational power of modern computers and with recent efficient algorithms, maintaining the full trajectory is possible with trivial compromises.

Eade and Drummond [2007] use a hierarchical bundle adjustment algorithm, in which multiple observations sharing a nearly-linear observation model are coalesced into nodes containing high-dimensional, rich observations, and the relations between these high-dimensional observations are optimised at the global level. Therefore optimising the linear parts of the parameter space proceeds recursively, permitting global optimisation at orders-of-magnitude less costly than bundle adjustment. This shows that it is possible to combine sub-maps generated through running a local Kalman filter to form locally linear nodes to form a globally optimal solution. When the camera moves to a new region, making the current filter estimate non-linear, the system marginalises out the latest camera pose to form a locally linear node, which contains only the set of landmarks initialised within that region. To make the estimates in the new region, a new Kalman filter is formed. Global constancy is maintained by estimating the inter-node similarity transformations using shared landmarks. This forms a graph to solve the SLAM problem.

Using SFM, Klein and Murray [2007] achieved real time performance for small augmented reality (AR) work spaces by splitting tracking and mapping into two separate tasks, processed in parallel threads on a dual-core computer, where one thread deals with the task of robustly tracking erratic hand-held motion, while the other produces a $3-D$ map of point features from previously observed video frames. The method adopts a stereo initialisation strategy, and occasionally makes use of local bundle updates for real-time performance on a local map. The method is different in that they attempt to build a long-term map in which features are constantly re-visited. They use two levels of bundle adjustment, one at local level and the other at global level when it is affordable. This is

the first implementation which can accommodate thousands of landmarks from the environment. De-coupling tracking from mapping enables this scalability. However, in order to preserve global consistency, doing a global bundle adjustment limits the scalable of the system.

Instead of solving the system of equations directly as in the conventional SFM approach, later work identified that there is a sparse structure in the Jacobian of the problem similar to that of the sparse structure of the information matrix in SEIF. In filtering, the weak links are created because of marginalisation, so the sparsity has to be enforced by removing those links. As SFM does not marginalise the poses, it naturally preserves a sparse pattern, as shown by Dellaert and Kaess [2006] in their square root filter for SLAM. This is a consequence of each camera pose in a SLAM system being connected only with a subset of landmarks, leading to a band-diagonal Jacobian structure. The square root filter solves the system of equations directly through elimination, without calculating the pseudo inverse as it has been done in the direct solvers, enabling efficient sparse matrix methods to be used for calculations.

Later, Klein and Murray [2007] showed how parameters can be appended incrementally into the system, which only changes a subset of parameters. This approach appends new parameters and then obtains the reduced form of the linear system through incremental QR decomposition. The fast incremental matrix factorization provides an efficient and exact solution.The key insight of this system is in showing that there is only a subset of parameters that needs updating in a SLAM system. Kaess et al. [2012] extend this idea to build a novel data structure, the Bayes tree, that provides an algorithmic foundation enabling a better understanding of existing graphical model inference algorithms and their connection to sparse matrix factorisation methods. Similar to a clique tree, a Bayes tree encodes a factored probability density, but unlike the clique tree it is directed and maps more naturally to the square root information matrix of the SLAM problem. Though incremental smoothing and mapping (ISAM) is quite efficient in solving exploratory problems, it could become less efficient when there are loops. Loops make optimisation less efficient as it connects more parameters

together. By using some heuristics for variable re-ordering it can be partially avoided (Kaess et al. [2012]).

Due to practical problems such as loop closure, SLAM systems can update only a subset of parameters within real-time constraints. Therefore, any sub-optimal solution will embed linearization errors into the system. This made it necessary to investigate more linear models. By representing all landmarks relative to its initialised camera frame Sibley et al. [2010] showed that the non-linearity of the model can be reduced significantly. They derived a new relative bundle adjustment framework, which instead of optimising in a single Euclidean space, works in a metric space defined by a connected Riemannian manifold. The fact that the variables in bundle adjustment are defined relative to a single coordinate frame has a large impact on the algorithms iterative convergence rate. This is especially true at loop closure, when large errors must propagate around the entire loop to correct for global errors that have accumulated along the path. In this approach, landmarks are represented relative to their initialised camera frame, allowing the frames to change minimally affecting the landmarks initialised on each frame.

Relative representation enables more efficient SLAM implementations, where only the most recent subset of parameters can be optimised instead of the whole system with minimal degradation to obtain comparable results with full bundle adjustment. In their work on double window optimisation for SLAM Strasdat et al. [2011] show how to take advantage of this linear relative representation. In order to achieve constant-time operation, the double window optimisation framework dynamically defines a sub-set of all key-frames as the active window over which to apply optimisation. In the optimisation the key-frames and points visible on those frames are included. Instead of bundle adjusting all frames and points, they define an outer window where points are marginalised out to yield constraints between frames. The resultant system is an approximation, because binary links between frames do not fully encode the nonlinear connections between frames and points. Nevertheless, it is capable of yielding reasonably accurate results. For better estimation accuracy, the front end of the system uses

dense image alignment. This system shows that with an accurate front end, it is possible to achieve near optimal results without full-bundle adjustment when implemented in a relative formulation.

Extending this notion Engel et al. [2013] track dense image patches with a non-negligible gradient to continuously estimate a semi-dense inverse depth map for the current frame, which in turn is used to track the motion of the camera using dense image alignment. Each estimate is represented as a Gaussian probability distribution over the inverse depth and this information is then propagated over time, and updated with new measurements as new images arrive. Even though global consistency is not explicitly maintained, the system yields accurate results, as the front-end tracker estimates the relative poses quite accurately through dense image alignment.

The large-scale direct monocular SLAM (LSD-SLAM) method by Engel et al. [2014a] not only locally tracks the motion of the camera but also allows building consistent, large-scale maps of the environment. The method uses direct image alignment coupled with filtering-based estimation of semi-dense depth maps. The global map is represented as a pose graph consisting of key-frames as vertices with $3-D$ similarity transforms as edges, elegantly incorporating the changing scale of the environment and allowing the detection and correction of accumulated drift.

Recent work on SLAM shows how accuracy improves with the information content used in the optimisation. This indeed is the reason for the current trend towards more denser systems. It is quite evident that this is the case as the underlying estimation problem is fully governed by the motion of the camera. This thesis investigates the dimensionality of the problem and how information flows from high-dimensional parameter space to update the camera pose.

2.2 Feature Matching and Pose Estimation

Image feature matching is an extremely important step in any SLAM system, as the accuracy of the final estimate is highly dependent on the matching accuracy. In monocular SLAM, when the system is initialised, correct matching is crucial to prevent the system deviating from the correct solution.

When the system is initialised with two or more camera frames, feature matches have to be used to solve for the relative pose between the frames and to recover the depth of each landmark in the environment. There are both closed form solutions as well as incremental methods to do this. Depending on the algorithm, a minimal number of matches are required to get the solution. If there are more matches than the minimal number required, a linear least squares minimization problem must be solved. When there are a large number of matches with outliers, random sample consensus (RANSAC) which iteratively generates a hypothesis of the parameters of a mathematical model from a set of observed data, can be used. Civera et al. [2009] introduce a method for embedding RANSAC withing the Kalman filter by temporarily updating the filter using a single randomly selected feature point to generate a hypothesis. Inliers of the hypothesis with the highest consensus score are then used to update the filter to obtain the final estimate.

One of the oldest and most common methods of pose estimation is to use 8-point correspondences. The classic paper by Longuet-Higgins [1987] shows how 8-points can be used for computing the essential-matrix. The method uses 8-points compute the structure of the scene from two views with calibrated cameras. If 8-point correspondences are known, the method involves solving 8 simultaneous equations for the essential matrix. The algorithm is very easy to implement and can be solved quickly. The main property of the method is the possibility of nicely encapsulating the epipolar geometry of the image configuration. If the two cameras used are uncalibrated, a set of equations for the fundamental-matrix can be solved as shown by Faugeras [1992] and Deriche et al. [1994]. However this

method is regarded as inferior to existing iterative algorithms as it shows a higher sensitivity to noise. Hartley [1997] later showed that the 8-point algorithm can yield solutions as good as any other pose estimation algorithm if the problem is conditioned properly. It is shown in this paper that a simple transformation (translation and scaling) of the points in the image before formulating the linear equations leads to great improvement and hence to the stability of the result.

When RANSAC is used, we can assume better performance if the number of points used in the hypothesis generation is small, as this reduces the probability of selecting outliers when generating the hypothesis. To this end it has been shown that a solution can be obtained, using either 7, 6 or 5 points. A good explanation of the 7-point algorithm can be found in the paper by Maybank et al. [1992]. The 6-point algorithm gives a unique solution and was presented by Philip [1996]. The 5-point algorithm proposed by Nistér [2004] consists of computing the coefficients of a tenth degree polynomial in closed form and subsequently finding its roots. This shows better numerical stability compared to the other methods.

Chapter 3

Corner Matching Refinement for Monocular Pose Estimation

Many tasks in computer vision rely on accurate detection and matching of visual landmarks (e.g. image corners) between two images. In particular, for the calculation of epipolar geometry from a minimal set of five correspondences, the spatial accuracy of matched landmarks is critical, because the result is very sensitive to errors.

The most common way of improving accuracy is to calculate a sub-pixel location independently for each landmark, in the hope that this reduces the reprojection error of the point in space to which they refer. In this chapter we present a method for refining the coordinates of *correspondences* directly. Given some coordinates in the first image, our goal is to maximise the accuracy of the estimate of the coordinates in the second image corresponding to the same real world point, without being too concerned about which real world point is being matched.

In this chapter we show how this can be achieved as a frequency domain optimization between two image patches to refine the correspondence by estimating affine parameters. We select the correct frequency range for optimization by identifying a direct relationship between the Gabor phase-based approach and the frequency response of a patch. Further, we show how parametric estimation can be made computationally efficient by operating in the frequency domain.

Finally, we present experiments which demonstrate the accuracy of this approach, its robustness to changes in scale and orientation, and its superior performance compared to other sub-pixel methods.

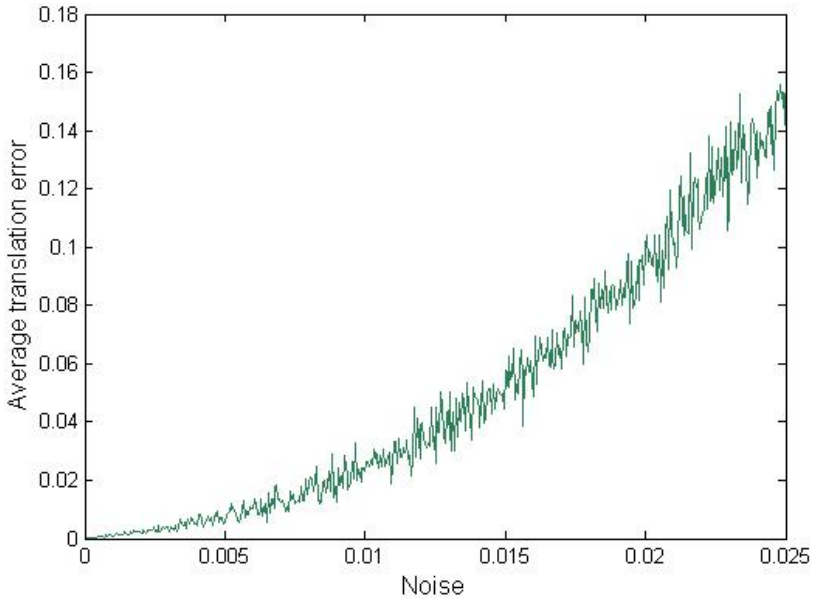
3.1 Introduction

Calculating the $3-D$ structure from an image sequence depends on accurately computing the motion of the camera. This in turn requires reliable feature extraction and matching. Therefore, as discussed in this chapter, accurate feature matching plays a very important role in accurate monocular pose estimation. A particular problem that drives this work is that of calculating the essential-matrix that describes the epipolar geometry of two images for which the internal camera parameters are known. This can be done from a minimal set of five correspondences between the two images using a polynomial solving algorithm, as described by Nistér [2004] which can generate up to ten essential matrices, or by iterative optimization of the residual error, as shown by Zhang et al. [2010]. In either case, the hypothesis generated from five matches is very sensitive to the accuracy with which the matches are extracted from the images.

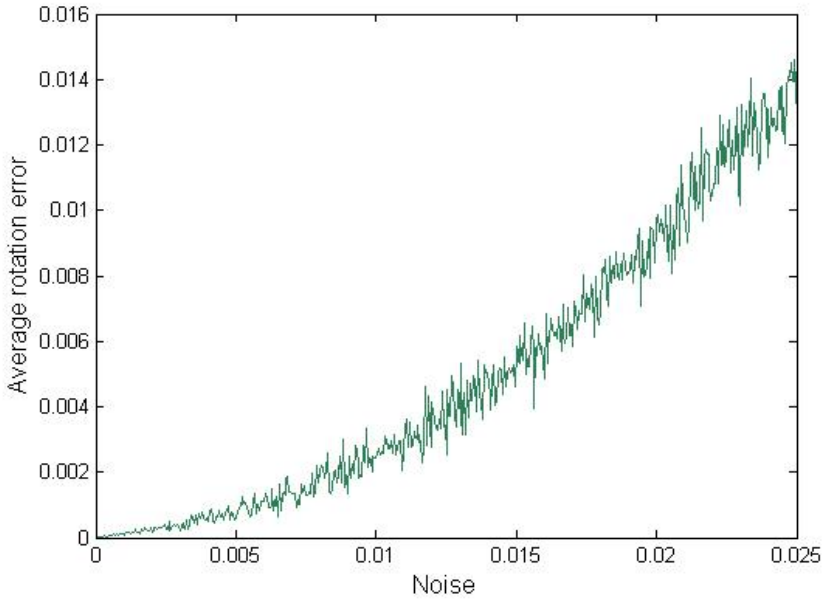
3.1.0.1 Monocular Pose Estimation

The sensitivity of essential-matrix calculation to errors in the data correspondences is partly a consequence of nonlinear error propagation with depth, which leads to a deviation from the desirable Gaussian uncertainty representation. In other words, depending on the depth to each physical point, the pixel error varies. Different parametrisation techniques have been proposed to reduce this non linearity to achieve better results, as shown by Montiel et al.. The inverse depth parametrisation is capable of greatly reducing the non-linearity of error propagation with depth. Although these methods are capable of making the pose estimation less sensitive to pixel noise, it still remains the major source that corrupts the final pose estimate, according to Civera et al. [2007].

To better explain the problem, a small experiment was conducted. In this experiment, 100 $3-D$ landmarks were randomly generated around a specified average depth (20 relative units) from the first camera and distributed uniformly. These points were then projected onto a second camera with a translation of 1 unit away from the first camera. An isotropic measurement noise was then added to the projected locations. These projected points with their original correspondences were then used to compute the least squares approximation to the essential-matrix between the two views from which the translation and the rotation were recovered and compared with the ground truth. It can be seen that both translation and rotation errors increase rapidly when the average noise level is increased. Figure 3.1 shows the results of a simulation experiment that illustrate this point. The graphs show how the translation and rotation error changes when the pixel errors increased gradually.



(a) Translation error versus noise



(b) Rotation error versus noise

Figure 3.1: Pose estimation error (estimated from two artificially projected camera frames) versus maximum noise magnitude

Most importantly, it can be seen that the translation estimate is more susceptible to pixel noise than the rotation. If the depth is further increased, the noise starts to dominate the estimated pose. For RANSAC based pose estimation, if the percentage of correct matches is relatively low, there is a very small chance of all five correspondences being selected to form the correct hypothesis in any iteration. This makes sub-pixel refinement extremely useful for accurate pose estimation, especially when the average scene depth is relatively large.

3.2 Previous Work

Most sub-pixel methods to date are for registering two images rather than for refining two feature matches. These assume a $2-D$ motion resulting from a motion in $3-D$ that can be described using a simple translation model. They have shown extremely high accuracy. Later some of these methods were extended to sub-pixel matching of two feature points, where usually one of the points in the first image is specified in advance and the corresponding point in the second image is searched. Widely used sub-pixel methods can be categorized as interpolation-based (correlation interpolation, intensity interpolation and geometric methods) as shown by Tian and Huhns [1986], Berenstein et al. [1987], phase correlation, and differential methods (optical flow and parameter optimization) as shown by Tian and Huhns [1986].

In the interpolation correlation method, sub-pixel registration accuracy is achieved through a discrete correlation function between two images and then fitting an interpolation surface to samples of this function. After this the maximum of this surface is searched accurately (Dvornychenko [1983], Anuta [1970]). The accuracy of interpolation-based methods depends on the quality of the used interpolation algorithm, but interpolation methods fail when handling projectively transformed images. On the other hand Foroosh et al. [2002] show that phase correlation can work well for sub-pixel registration. Conventional phase corre-

lation techniques fail when the matching window under consideration becomes relatively small. Recent work by Shibahara et al. [2007] has shown the necessity to fit a function to the phase correlation measurement in order to obtain satisfactory results under such constraints. Although both of these methods can be extended to sub-pixel patch matching/refinement, their applicability is limited to simple translation models, and any affine transformation needs to be rectified separately at a prior stage. Differential methods use a constraint equation under intensity conservation assumption, as shown by many authors, including David [1987], Berthold and Brian [1981] and Nagel or handle the problem as an optimization problem over a set of parameters, as shown by Bergen et al. [1992], which works well under local patch deformations.

Recently considerable attention has been given to more complex models of motion with a set of parameters because of their ability to provide a good approximation to local $2-D$ motions. Because of their noise sensitivity and better convergence, parametric motion models have been extended to the frequency domain in the work of Kruger and Calway [1996]. These methods use the shift invariance of the magnitude spectra to separate the translation component from the other four affine parameters. The translation is estimated using the phase correlation between two affine-rectified images.

In the frequency domain, the phase of a signal bears most of the information compared to the magnitude spectra as it has been shown by Hayes et al. [1980]. On the other hand, the phase has shown much robustness to noise, as Fleet and Jepson [1993] show. According to this insight, it is possible to obtain better results if phase is not discarded and is used in the optimization equation by optimising all six affine parameters simultaneously.

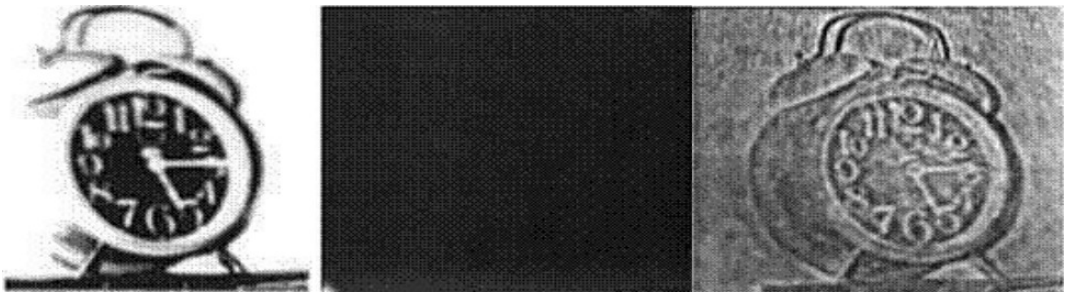


Figure 3.2: Signal synthesis from magnitude and phase spectra respectively.

Figure 3.2 shows a signal synthesised from the magnitude and the phase spectra. When the original image signal on the left-hand side is re-synthesised from the magnitude and the phase spectra respectively, more information can be recovered from the phase than the magnitude. The frequency response of a signal can be analysed from the Fourier transform.

To improve the accuracy of the generated hypothesis, descriptor-based feature matching which works at pixel level may not be adequate, and sub-pixel level information may be needed. A number of applications already use something better than pixel-sampled information. Sub-pixel methods have been used extensively for stereo matching (Scharstein et al. [2001], Matthies et al. [1989]). However most of these techniques are based on the assumption that the $2-D$ image motion, resulting from $3-D$ camera motion can be described using a simple translation model (Donate et al. [2011]). Widely used sub-pixel methods are interpolation-based (correlation interpolation, intensity interpolation and geometric methods) (Berenstein et al. [1987]), phase correlation and differential methods (optical flow and parameter optimization) (Tian and Huhns [1986]).

Phase correlation works well for sub-pixel registration, but conventional phase correlation techniques fail when the matching window under consideration becomes small. Recent work has shown the necessity to fit a function to the phase correlation measurement in order to obtain satisfactory results under such

constraints (Shibahara et al. [2007]). Although these methods can be extended to sub-pixel patch matching/refinement, their applicability is limited to simple translations, and any affine transformation needs to be rectified separately at a prior stage. Differential methods use a constraint equation under intensity conservation assumption (David [1987], Berthold and Brian [1981], Nagel) or handle the problem as an optimization over a set of parameters (Bergen et al. [1992]), which works well under local patch deformations. Therefore, in recent years considerable attention has been given to more complex motion models based on parameter estimation (Campani and Verri [1992]). Such methods are based on hierarchical or multi-resolution approaches limit their applicability in time-critical applications. Because of its noise sensitivity and better convergence, the parametric motion model has been extended to the frequency domain (Hsu et al. [1993]; Kruger and Calway [1996]). Such frequency domain approaches have shown better performance and noise tolerance compared to spatial domain methods. These methods use the shift invariance of the magnitude spectra to first estimate the four non-translation affine parameters. The translation is then estimated using phase correlation between affine-rectified images.

In the frequency domain, the phase of each frequency contains much more information than the magnitude (Duan and Robert [1989], Hayes et al. [1980]), and shows better robustness to noise (Fleet and Jepson [1993]). By not discarding phase, and simultaneously optimising all six parameters it is possible to obtain better results. We parametrise the signal using the six-parameter affine model with an additional parameter to compensate for energy changes of the signal. By doing the optimisation in the frequency domain, it is possible to achieve improved results and a fast convergence rate.

3.3 Contributions

In this work we develop a frequency domain Gauss Newton optimization framework for sub-pixel image alignment. This work is completely novel as we draw a relationship between the conventional Gabor filter and the frequency response of a Gaussian multiplied image patch to do the sub-pixel alignment in the frequency domain. The main contributions are as follows:

- For sub-pixel refinement, we represent the local affine transformations in the frequency domain and optimise over all affine parameters simultaneously, using both magnitude and the phase information.
- We model the local transformations of a projectively transformed image pair by an affine transformation, selecting a 32x32 patch around two corresponding corners and try to refine the second corner position by affine warping the frequency spectrum of the first patch and changing the phase of the second.
- In order to further increase accuracy, we re-sample the second patch using the estimated translation.
- We derive a relationship between the Gabor filter phase difference and the frequency representation of a Gaussian weighted image patch and use this to select the effective frequency range for optimization.
- Using several sub-sampling stages in the frequency and spatial domains, we obtain better sub-pixel accuracies (down to 0.1 pixels under moderate affine transformations) and better convergence.
- These sub-pixel refined correspondences are then used to achieve a more stable and accurate pose estimate.

3.4 Gabor Filter

In his seminal work, starting from the uncertainty principal Gabor [1946] derived that a Sin-modulated Gaussian patch can act as the signal that can be represented covering a smallest area in the time-frequency plane. The Sin-modulated Gaussian signal later used as a filter known as the Gabor filter has many applications in wavelet analysis.

Two properties of the Gabor filter have to be tuned, depending on the application. These are the variance of the Gaussian kernel and the frequency of the modulated sinusoid. The frequency of the modulation controls the precision of the filter. This frequency is known as the principal frequency of the filter. A complex Gabor filter is defined as the product of a Gaussian kernel times a complex sinusoid, i.e:

$$Gabor\left(x;\sigma,k_0\right) = e^{ixk_0}G\left(x;\sigma\right) \tag{3.1}$$

Usually, the standard deviation σ of the Gaussian distribution is chosen such that it removes the DC response of the filter, which is known as the one octave bandwidth of the filter, as shown in Section 3.5.2. What is important is the frequency response of this filter. In the frequency domain, the Gaussian function remains a Gaussian. The effect of the frequency modulation is to shift it in the frequency domain by the amount specified by the principal frequency of the modulation. Therefore, it is equivalent to analysing a particular portion of the signal in the frequency domain around the principal frequency.

In image processing, the Gabor filter, when defined as a function of the spatial location of the image and the frequency of the filter, is known as the Gabor scale space, where the standard deviation is selected with respect to the tuning or the

principal frequency such that:

$$S(x, \lambda) = Gabor(x; \sigma(\lambda), k(\lambda)) * I(x) \tag{3.2}$$

where, $*$ is the convolution operator and λ is the scale parameter. Here, since we are not changing the scale of the filter the scaling parameter remains constant. As it can be assumed that the scale change of SLAM is comparatively small, the effect of neglecting the scale difference is minor. In the next section we give the mathematical derivation of our method in detail.

3.5 Estimation of Affine Transformations

In this section we provide the mathematical derivation of the frequency domain affine parameter estimation. First we start with an intuitive explanation of the method that has been developed.

As explained earlier, the Gabor filtering of an image is the convolution of the image with a Sin-modulated Gaussian. This makes the frequency response of the filter another Guassian, shifted by the modulated frequency. Lower the frequency, higher will be the precision of the filter in identifying more subtle variations of the image signal. However, the filter cannot start with a very small frequency, as this would lead to aliasing, making the response useless, as shown in Section 3.5.2. As a result, in practice, a Gabor filter bank is usually used with varying frequencies ranging from a lower frequency to a higher frequency. As the image is $2-D$ the filter has to be directionally tuned, as describe by Fleet and Jepson [1990] to obtain the response along a particular direction, as shown in Figure 3.5.

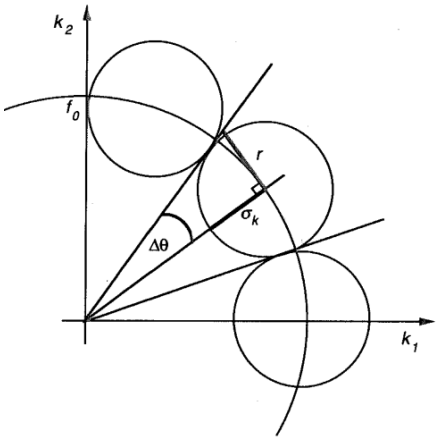


Figure 3.3: Directionally tuned Gabor filters.

Image alignment using a Gabor filter bank is expensive, as it is necessary to obtain those responses in each direction. In addition, it is not easy to so finely sample all directions around a given point in the image. Here we note that the responses for the given range of frequencies of an image patch can be directly obtained using the convolution theorem.

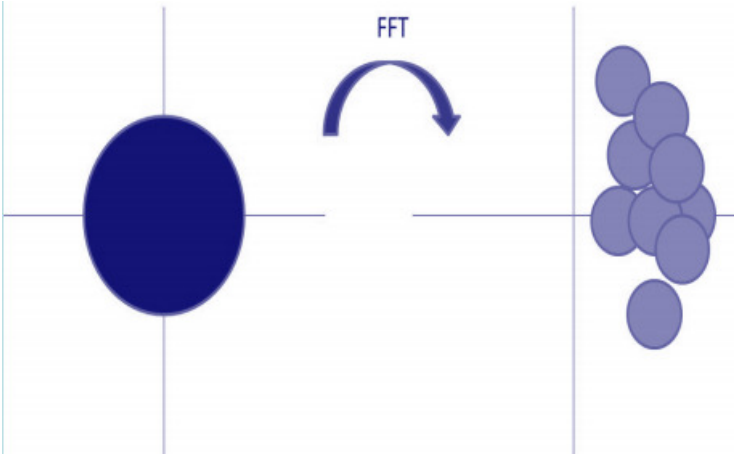


Figure 3.4: Frequency response of a Gaussian multiplied image patch.

Consider a Gaussian multiplied image patch. As can be seen from Figure 3.5, the frequency response of a Gaussian multiplied image patch becomes the convolution between the frequency response of the image patch and the Gaussian kernel in the frequency domain. Physically what has happened is, the Fourier transformation is shifting the Gaussian signal in the frequency domain over the frequency response of the image patch to generate responses at each frequency.

If we take a particular point in the frequency domain of this convolution response, it can be seen that it is nothing other than a Gabor filter response which has the same Gaussian envelope as the original Gaussian, but modulated by a sinusoid with the same tuning frequency as that of the selected point. This observation indicates that it is possible to obtain the responses of a Gabor filter bank directly by reading a particular point on the frequency domain of a Gaussian multiplied patch. The tuning frequency is the frequency at that selected frequency location and the direction of the filter is the selected direction in the frequency domain. We use this property to derive a frequency domain Gauss-Newton optimization framework.

To present the affine parameter model in the frequency domain, we make use of the affine theorem in frequency (Bracewell et al. [1993]). Given two image patches $I_0(\bar{x})$ and $I_1(\bar{x})$ surrounding two corresponding corners, which are related by an affine coordinate transformation $I_1(\bar{X}) = I_0\left[A^{-1}\left(\bar{X} - \bar{b}\right)\right]$, their $2-D$ Fourier transforms are related by:

$$\hat{I}_1(\bar{u}) = |det\,(A)|\, e^{-j\bar{u}.\bar{b}} \hat{I}_0\left(A^T\bar{u}\right) \tag{3.3}$$

The shift invariance property of the magnitude spectra of Equation 3.3 enables the estimation of $\bar{b}$ to be separated from the estimation of A (Kruger and Calway [1996]). However, discarding phase information is a great waste, as the phase carries a good deal of information in the frequency domain, which can be used to achieve more stable and fast estimations by simultaneously optimising all six

parameters.

Here we use the six parameter affine model with an additional parameter. The seventh parameter compensates for energy changes caused by different local illumination conditions. If we select $\bar{\beta} = \{\beta_1 \dots \beta_7\}$ to be the parameter set and absorb the $|det\,(A)|$ of Equation 3.3 into β_7 we have:

$$\beta_7 \hat{I}_1(\bar{u}) = e^{-j\bar{u}\cdot\bar{b}} \hat{I}_0\left(A^T \bar{u}\right) \qquad \text{where} \quad A = \begin{pmatrix} \beta_1 & \beta_2 \\ \beta_3 & \beta_4 \end{pmatrix} \quad \text{and} \quad \bar{b} = \begin{pmatrix} \beta_5 \\ \beta_6 \end{pmatrix} \tag{3.4}$$

$$\tag{3.5}$$

This can be rearranged to yield:

$$\beta_7 e^{j\bar{u}\cdot\bar{b}} \hat{I}_1\left(\bar{u}\right) = \hat{I}_0\left(A^T \bar{u}\right) \tag{3.6}$$

Thus, the error r, for a frequency $\bar{u}$ can be written as:

$$r\left(\bar{u}, \bar{\beta}\right) = \beta_7 e^{j\bar{u}\cdot\bar{b}} \hat{I}_1\left(\bar{u}\right) - \hat{I}_0\left(A^T \bar{u}\right) \tag{3.7}$$

The above equation enables us to model the affine transformation as a phase change of $\hat{I}_1$ and a warp of $\hat{I}_0$ with respect to matrix A. The Jacobian J_i of the partial derivatives of r with respect to β_i can then be computed easily using the chain rule. The computed full Jacobian then becomes:

$$[J_1,\ J_2,\ J_3,\ J_4,\ J_5,\ J_6,\ J_7] = \left[-\frac{\partial I_0}{\partial u}u,\ -\frac{\partial I_0}{\partial u}v,\ -\frac{\partial I_0}{\partial v}u,\ -\frac{\partial I_0}{\partial u}v,\ -\beta_7\tilde{I}_1 u,\ -\beta_7\tilde{I}_1 v,\ \tilde{I}_1\right] \tag{3.8}$$

Given a set of frequencies $\{u_j\}$, the errors $r(u_j)$ and the Jacobian J_{ij} can be used to obtain the parameters $\bar{\beta}$ that minimise $E = \sum_j \|r(u_j)\|^2$ using the Gauss-Newton method.

3.5.1 Iterative Refinement

After initialising the set of parameters by setting A to be the identity matrix and $\bar{b}$ to a zero vector, we use the Gauss-Newton method to warp the frequency patch $\hat{I}_0$ with respect to the first four parameters $\beta_1 \dots \beta_4$ and phase shift the patch $\hat{I}_1$ with the remaining two parameters β_5 and β_6.

Warping is done by sub-sampling the original frequency patch using bi-linear interpolation. After optimising for two or three iterations in the frequency domain, we extract the parameters β_5 and β_6, which correspond to a translation in the spatial domain in x and y directions, respectively. These two parameters are then used to re-sample the second patch (patch I_1) in the spatial domain at the new refined position using spatial sub-sampling. The Fourier transform of this re-sampled patch is then used to re-estimate a new set of affine parameters. This routine is continued until sufficient accuracy is achieved. According to experimental results, two spatial sampling steps are usually sufficient to reduce the average pixel error down 0.1 pixels.

3.5.2 Aliasing and the DC Response

Two issues need to be addressed in order for the frequency domain optimization to work in practice. Firstly, the method as presented is sensitive to edge effects on the border of the image patch. Secondly, the presence of a large DC component in the signal causes issues at low frequencies (those where $\|u\|$ is small).

In order to remove edge effects from an image patch, we multiply the patch by a Gaussian weighted window $G(x, y)$ centred on the detected landmark before computing the Fourier transform. The patch also contains a large DC component, which appears as a large spike at $u = 0$ in the frequency domain. The frequency domain convolution that results from Gaussian windowing spreads this out along small frequencies. To overcome this issue, the DC component of the Gaussian windowed patch is removed, yielding a new patch I' defined as:

$$I'(x, y) = G(x, y) \left(I(x, y) - \frac{\sum_{x,y} G(x, y) I(x, y)}{\sum_{x,y} G(x, y)} \right) \tag{3.9}$$

This results in a patch with 0 DC coefficient that fades towards 0 at the boundaries. The frequency response of the Gaussian multiplied patch, $\mathcal{F}[I']$, has a direct relationship with the Gabor filter with an identical Gaussian support. We use this relationship to select the useful frequency range (in order to eliminate possible aliasing effects) for the optimization in a multi-resolution manner.

Multiplying the patch by a Gaussian in the spatial domain is equivalent to convolving it by a Gaussian in the frequency domain, as explained earlier. This turns the values of the Fourier transform of I' into responses of the original patch (with applied DC offset) to complex Gabor filters. This interpretation can be used to select the useful frequency range for the optimization.

Because the phase of a particular Gabor filter response changes linearly under spatial translations of the signal, this has been used for spatial disparity estimations (Fleet et al.). The phase disparity is useful only if the displacement is smaller than a half a wavelength of the tuning frequency (Fleet et al.), i.e the domain of convergence for the phase is $\pm\pi$ as shown in Figure 3.5.2.

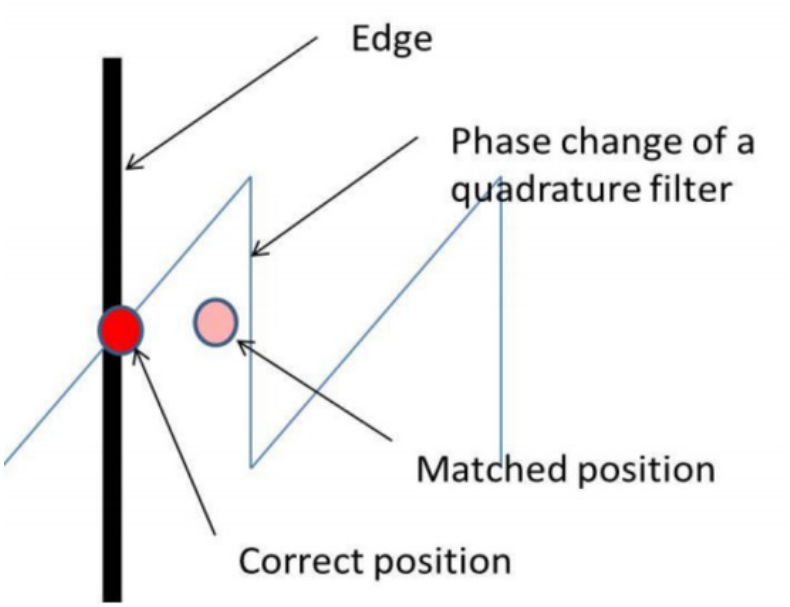


Figure 3.5: Aliasing in the frequency domain.

This imposes an upper limit over the useful frequency range. If we assume a maximum displacement of d pixels for a $1-D$ signal, this criterion suggests a frequency f such that $f \leq 1/2d$. In the $2-D$ case we can meet this criterion by limiting the useful frequency range radially to a maximum of $1/2d$ radius. After estimating the translation (and other parameters) using small frequencies for large displacements, this can then be refined by gradually increasing the radius, incorporating higher frequency responses in the optimization.

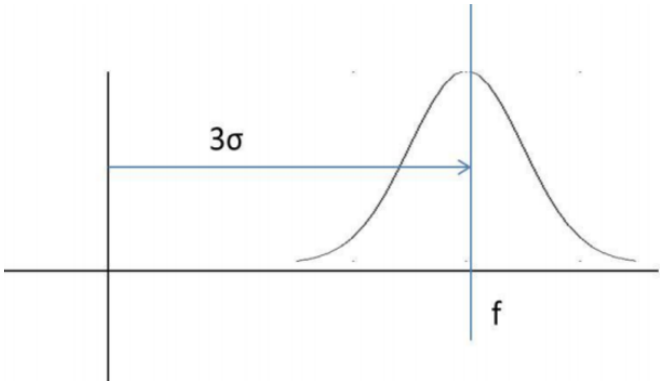


Figure 3.6: One octave bandwidth to eliminate the DC response.

Although, spatial subtraction in Equation 3.9 can greatly reduce the signal being corrupted by the DC component, for better results we have to impose a lower frequency limit. We select the minimum frequency using the one octave bandwidth criterion 3.6, as suggested in the literature (Fleet and Jepson [1993]) for Gabor filter-based disparity estimations. The one octave bandwidth in the frequency domain for the Gabor filter shows that, to eliminate the effect of the DC part of the signal, the spatial support is:

$$\sigma = \frac{1}{2\pi f}\left(\frac{2^{\alpha}+1}{2^{\alpha}-1}\right) \tag{3.10}$$

If we select the frequency f keeping the spatial support σ constant, in order to eliminate any DC distortion, the minimum frequency should be:

$$f \geq \frac{1}{2\pi\sigma}\left(\frac{2^{\alpha}+1}{2^{\alpha}-1}\right) \tag{3.11}$$

Combining the minimum and the maximum criteria for frequency selection gives the useful frequency as follows:

$$\frac{1}{2d} \geq f \geq \frac{1}{2\pi\sigma}\left(\frac{2^{\alpha}+1}{2^{\alpha}-1}\right) \tag{3.12}$$

At the end of each iteration we can expect the displacement d to reduce, which in turn expanding the useful frequency range without aliasing. Higher frequencies carry finer details about the translation, which improves the resolution of the final estimate. This naturally enables a multi-resolution framework for refinement without any additional computations.

Figure 3.7 summarises the steps we use to sub-pixel refine a target corner position with respect to the given reference corner.

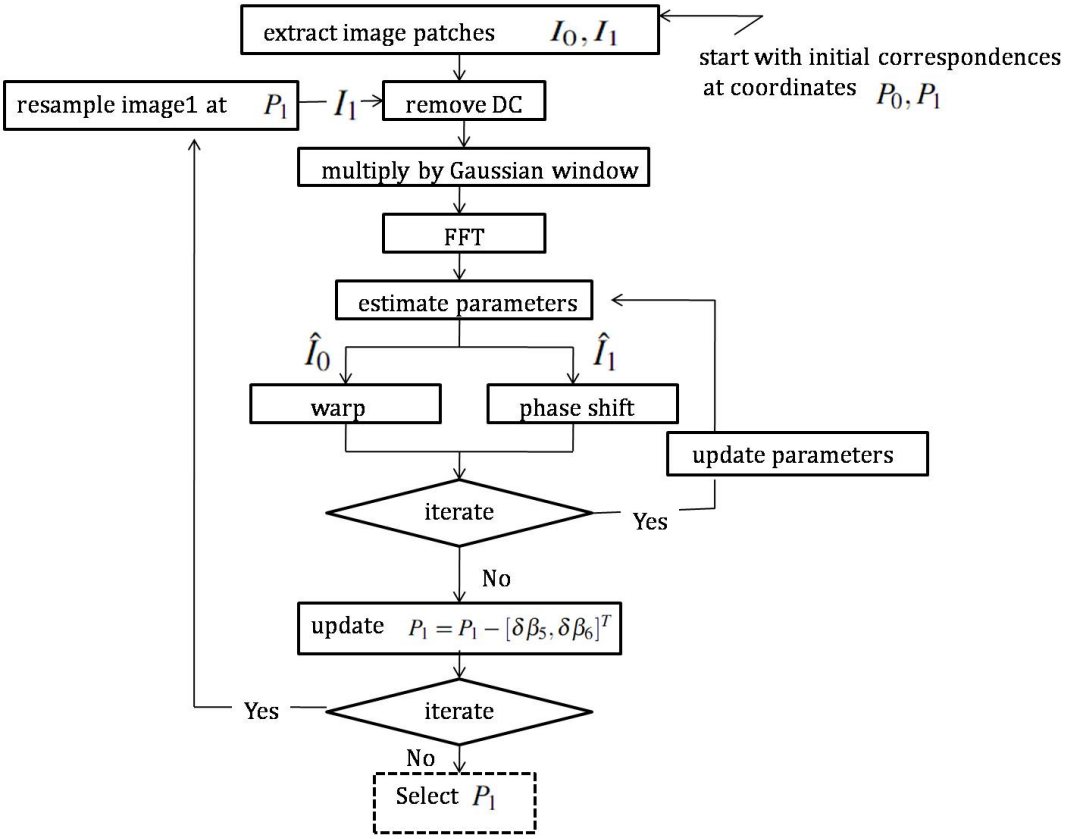


Figure 3.7: Iterative optimization based on sub-sampling in the frequency and the spatial domains.

3.6 Experiments

In this section we apply the proposed refinement method to refine corner correspondences and compare the results with the results of the spatial domain Gauss-Newton counterpart. The spatial method performs an iterative Gauss-Newton optimisation over all six affine parameters to minimise the sum of the squared differences between I_0 and the affine transformed I_1, sampled with bi-linear interpolation such that:

$$\sum_{\bar{X}} I_1(\bar{X}) - I_0\left[A^{-1}\left(\bar{X} - \bar{b}\right)\right] \tag{3.13}$$

In order to compare the two methods, here we apply them to sub-pixel refine the matches between synthetically generated images and use the refined matches for parameter estimation. First, we estimate the affine transformation between two images after sub-pixel refinement. Next, homographies are estimated and finally it is used to estimate the epipolar geometry.

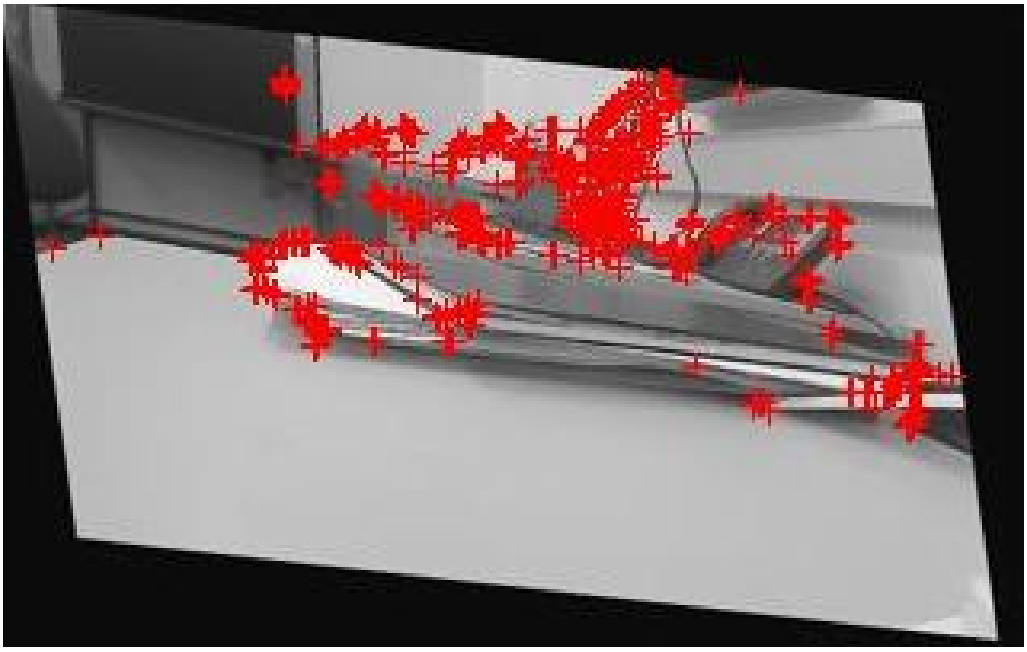
3.6.1 Synthetically Generated Transformations

We first perform an experiment to demonstrate the improvement that can be achieved through refinement in a situation where the ground truth transformation is known. Synthetic data is generated first by transforming a reference image using a known affine transformation through bi-linear interpolation. To remove the interpolation artifacts, both images in the pair are down-scaled by a factor of two. Then FAST features are extracted from the first image and projected onto the second, using the same transformation.

The projected corners are rounded-off to the nearest integer pixel. The raw pixel errors are calculated as the distance between the ground truth and the rounded-off positions. Figure 3.9 shows the typical performance.

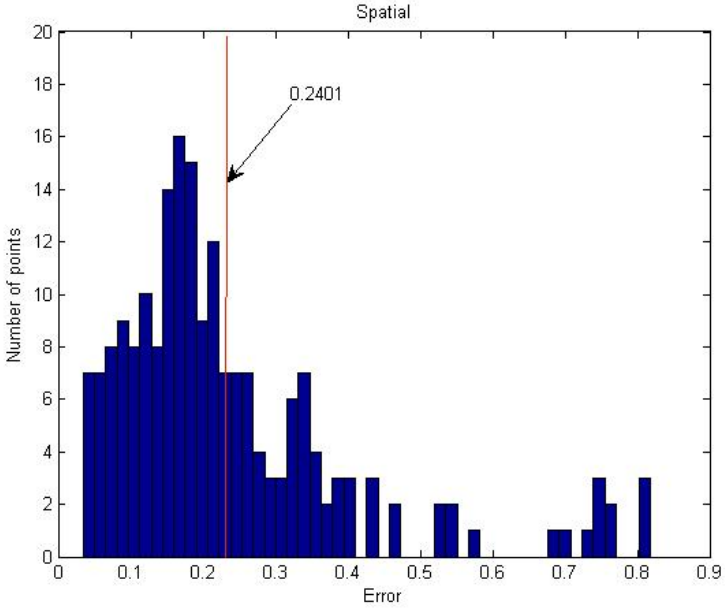
Here, the average unrefined pixel error is 0.4102. The two methods are then used to refine the coordinates of a match in the second image. The spatial method reduces the average error to 0.2423, while the frequency-based method reduces it further to 0.1454.

(a) Original image

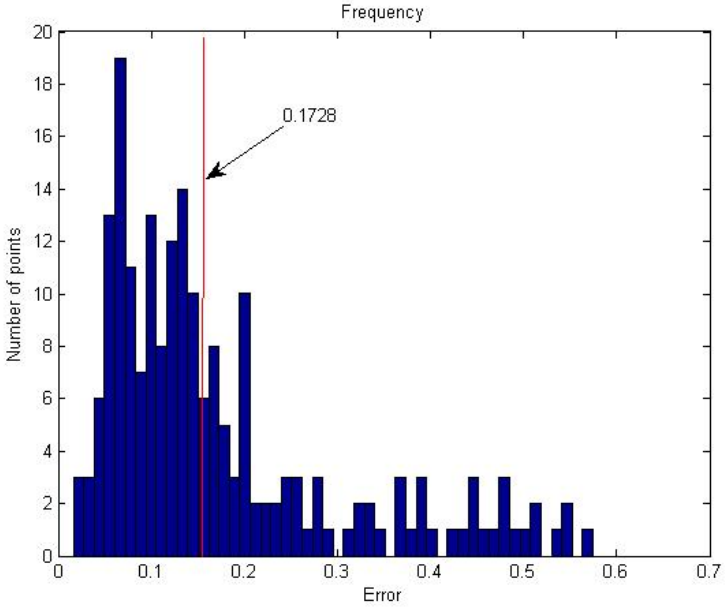


(b) Transformed image

Figure 3.8: Reference image pairs.



(a) Error distribution after spatial domain refinement



(b) Error distribution after frequency domain refinement

Figure 3.9: The error distributions.

3.6.2 Homography Estimation

Here, two images of the Graffiti database shown in Figure 3.6.2 were used. Then the FAST features were extracted from both images and were using histogram intensity patches (HIPS Taylor and Drummond). Then the inliers were selected by applying RANSAC to these raw matches and kept fixed. The inliers were then refined using both spatial and frequency methods.

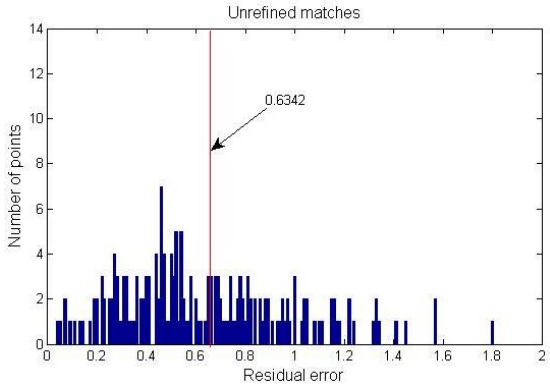
The matches were then used to estimate the homography, first with raw matches, then spatially refined matches, and finally the frequency refined matches minimising $\sum_i r_i^2$ where $r_i = \sqrt{\|x_i - Hy_i\|^2 + \|y_i - H^{-1}x_i\|^2}/2$. Each x_i is a homogeneous image coordinate of a FAST feature in image 1 and y_i is the corresponding homogeneous image coordinate of the refined location in image 2. Figure 3.11 shows the distribution of the residual errors of each of the three schemes (raw, spatial and frequency) applied to this image pair. The average raw errors are as follows:

- Unrefined - 0.6342
- Spatial domain refinement - 0.3464
- Frequency domain refinement - to 0.2285.

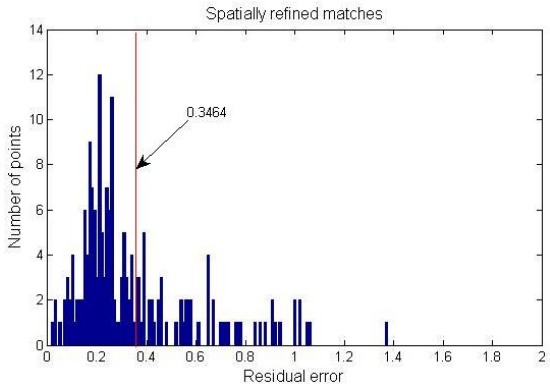
(a) Original Image

(b) Synthetically transformed image

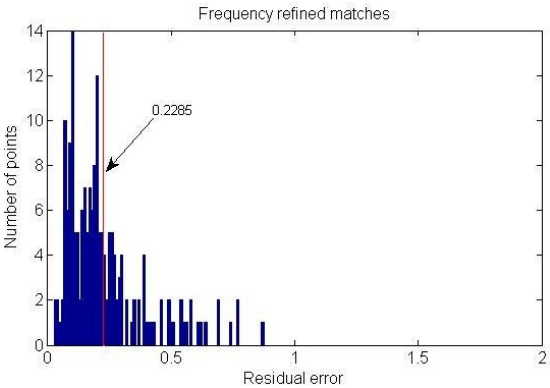
Figure 3.10: Original and transformed images.



(a) Unrefined



(b) Spatial



(c) Frequency

Figure 3.11: Residual error distribution for the first image pair in Graffiti database

3.6.3 Pose Estimation

As discussed in Section 3.1.0.1, pose estimation from an essential-matrix is extremely sensitive to matching errors, and benefits from sub-pixel refinement. For a set of image pairs, we estimated essential-matrices using the five-point pose algorithm for FAST corner matches. Then the symmetric point to line error was calculated for each correspondence. The matches were then refined using the frequency domain and the spatial domain methods and the essential-matrix was re-estimated. Further, we compared the results with the sub-pixel coordinate estimation of SIFT difference of Gaussian features matches. Table 3.1 summarises those errors. Figure 3.12 shows two of the image pairs used for the pose estimation.

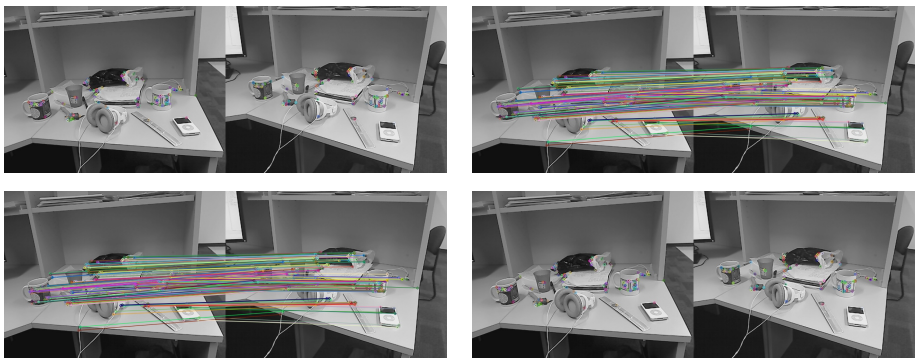


Figure 3.12: Two image pairs used for pose estimation

		Residual error in pixels for image pairs		
		P1	P2	P3
HIPS	Raw	0.447879	0.571511	0.535419
	Refined	0.360749	0.384233	0.373508
Sub-pixel SIFT		0.541322	0.435442	0.690141

Table 3.1: Comparison of frequency-based refinement with sub-pixel features from SIFT.

3.7 Discussion

In this work we have extended the affine parameter estimation for match refinement that operates in the frequency domain. Importantly, we have shown the ability of the newly proposed method to refine correspondences in a coarse-to-fine multi-resolution manner in the Fourier domain.

Experimental results establish the effectiveness of the proposed method for modeling local patch deformations which can be used for sub-pixel refinement. Such locally refined corners are then used to estimate the global monocular pose with improved accuracy. As a post-processing step, after a less accurate but fast descriptor-based feature-matching stage, our method can be used for efficient sparse match refinement.

However, due to the fixed size of the Gaussian weighting function, we found that the refinement accuracy is sensitive to scale changes if it is more than $20-30\%$. If the image pair contain scale changes larger than this, it is necessary to use image pyramids with at least two layers per octave. Further, if the translation is large compared to the minimum half wave-length of the selected frequency band, the solution degenerates as the Hessian matrix in Gauss-Newton algorithm becomes ill-conditioned. Therefore, the accuracy of the final result depends on the coarse-to-fine frequency tuning of the optimization. Fortunately, in practice, the feature detection and matching methods used in this chapter give correspondences that are well within the convergence band of our algorithm.

With front end discussed in this section, next we introduce a dimensionality reduction technique for EKF-SLAM in the next chapter. We follow the conventional EKF-SLAM formulation where we keep only the current camera pose in the filter by marginalizing out all previous camera poses.

Chapter 4

Reduced Dimensionality Extended Kalman Filter for SLAM

The computational complexity of the Kalman filter grows at least quadratically with the number of dimensions in the filter. This is a particular problem for applications like SLAM, where it is not possible to run a single filter on a large-map with many thousands of landmarks.

In this chapter we present a method for dramatically reducing the computational complexity of the Kalman filters by reducing the dimensionality as information is acquired. We apply it to monocular SLAM, where there is a large number of dimensions in the filter that are not subject to process noise (the landmark locations). This has the effect of reducing the cost of running a filter or allowing a single filter to process a much larger set of landmarks.

Our approach also has a role to play within modern efficient sparse matrix approaches for SLAM where local information is coalesced into key-frames using Kalman filters. It also has general applicability to filtered measurement of static quantities where there are large numbers of dimensions that are not subject to process noise.

Despite the fact that there are efficient sparse matrix methods for SLAM, we propose the method in this chapter as an alternative filter-based approach which will be useful as a local solution to a large graph-based SLAM system or any Kalman filter-based application with dimensions that are not subject to process noise in general. In the next chapter we show how to extend this to much larger scales using a relative formulation.

4.1 Introduction

A SLAM system continuously explores the environment to causally estimate the ego-motion of a robot and to map the environment. Many successful SLAM algorithms represent the stochastic nature of the robot motion and the measurement models, together with noisy sensor data in a probabilistic manner, tracking the joint posteriori over the vehicle pose and the map.

The filtering approach for SLAM maintains only the current camera pose with all landmarks of interest as the state (Davison et al. [2007], Azarbayejani and Pentland [1995], Jin et al. [2003], Chiuso et al. [2002]). This representation marginalises previous camera poses, which in turn connect all state elements with each other, making the covariance matrix dense, leading to a fully connected graph. With new observations, the filter complexity grows at least quadratically, quickly making the problem intractable. This quadratic growth can be handled by dividing the whole problem into small tractable maps while separately maintaining global consistency (Eade and Drummond [2007]). Developments like parallel

tracking and mapping (PTAM) (Klein and Murray [2007]) and dense tracking and mapping (DTAM) (Newcombe et al. [2011]) take this further by noticing the possibility of working with an active set of landmarks by separating tracking from mapping. In these systems the tracker, works with a local set of landmarks while the map maintains global consistency through bundle adjustment. Full bundle adjustment in PTAM adjusts the pose with respect to all key-frames. It exploits the sparseness inherent in the structure-from-motion problem to reduce the complexity. Without maintaining all landmark descriptors in this manner one could even use a more efficient sparse matrix system (Thrun et al., Walter et al. [2007], Dellaert and Kaess [2006], Kaess et al. [2008], Kaess et al. [2012]) as the back end to build a globally consistent map. As efficient as sparse matrix methods are, they still have limitations and are not used to process all frames of video as this would generate much denser graphs with high connectivity, which would overwhelm the approaches. Hence, they are restricted to using sparsely sampled key-frames.

On the other hand, coalescing observations into independent local key-frames by building a graph of local nodes, and optimising the resulting graph (Davison et al. [2007]) effectively overcomes the problem with key-frames in order to maximise the amount of information available. Because the information is acquired locally, the problem is nearly linear, and therefore a Kalman filter was used to combine information from multiple frames to give rich key-frames that know something about the inverse depth of landmarks as well as their image location. Hence, the Kalman filter still has a role to play.

In this chapter we present a method to reduce the dimensionality of the extended Kalman filter (EKF) for SLAM by identifying dominant modes of the filter. The method proposed can be used in general to reduce the dimensionality of the EKF irrespective of its application, without being limited to SLAM. We are particularly interested in reducing the dimensionality of each node of the SLAM system mentioned in the previous paragraph (Eade and Drummond [2007]), as this graph optimization problem is nearly linear with the number of nodes and as it combines the information from multiple frames to give richer key-frames.

The most complex operation in such systems is the Kalman filtering within each node, which imposes an upper limit on the maximum number of landmarks each node can handle. Our approach can reduce the dimensionality of Kalman filters used in each node, increasing the number of features a node can handle, leading to a more accurate estimation (Clemente et al. [2007]). As we suggest in the discussion such a dimensionality reduced graph can even be used in a multi-camera set-up more efficiently than a key-frame based sparse matrix method.

Here we consider the particular class of Kalman filters where the process noise is zero for a large number of dimensions. To handle large fixed-size data sets which change dynamically, a reduced Kalman filter (Farrell and Ioannou [2001]) is used in weather forecasting. Localization and mapping differ from this, as the state vector size grows with time. In addition in SLAM there is a large number of dimensions with zero process noise. If the process noise is zero, the filter's n dimensional state vector can be regarded as a point in a $n - D$ space, with a hyper ellipse representing its uncertainty, given by the covariance matrix. After several observations, the uncertainty in some directions in this space will reduce, making further information along the same directions obsolete. This implies that information has to be fused only along other directions where enough evidence is not yet available. In other words, observations contain useful information only along directions we are very uncertain about. Using these principal modes we reduce the dimensionality of the EKF and maintain a reduced state It should be noted that our method is fundamentally *different* from *sub-mapping* approaches, as we are removing only less uncertain directions of the filter, indirectly maintaining all features of the filter in contrast to removing inactive landmarks as is done in sub-mapping.

We re-parametrise the filter in terms of uncertainty and work with a reduced number of uncertain dimensions, maintaining a covariance matrix only in this reduced space (see Section 4.5). This re-parametrization takes place repeatedly whenever the filter has enough information about a degree of freedom (it has a sufficiently small singular value) such that it can freeze that dimension and remove it from the filter (see Section 4.4.2). This process frees up computational

capability, which in turn we can use by adding more state variables (corresponding to the locations of additional landmarks) to the filter (see Section 4.4.1). Camera pose (and possibly velocity) in the EKF are subject to process noise, hence are not suitable for dimensionality reduction because the process noise introduces uncertainty in those variables. In Section 4.5.1 we show how such variables can be handled within our reduced framework. Section 4.6 gives a complexity analysis of the system and Section 4.7 provides experimental results. Then we conclude (see Section 4.8) by explaining how our method can be used for multiple camera SLAM.

4.2 Background

In the past, the EKF has been successfully employed by many structure from motions (SFM) algorithms (Azarbayejani and Pentland [1995], Jin et al. [2003]). Causal integration or recursive estimation of SFM has enabled highly accurate real-time reconstruction of the environment while localising the camera (Chiuso et al. [2002]). The main reason for the success of recursive SFM algorithms is the possibility of capturing many features for motion estimation. The feasibility of EKF for real-time monocular SLAM was demonstrated first by Davison [2003]. The EKF maintains a full covariance matrix of size $n \times n$ for n landmarks. In this setting, the computational cost for updating the state becomes $\mathcal{O}(n^2)$ for each observation, making real-time operation infeasible if the number of landmarks exceeds more than 100. For this reason the operation of the system has to be limited to room-size environments (Eade and Drummond [2006]).

To overcome the scaling problem, extensive research has been undertaken by the robotics community (Bosse et al. [2003], Montemerlo et al. [2002], Leonard and Feder [2001]). Dividing the environment into sub-regions with a manageable number of features is one way of handling the complexity of the problem. A separate EKF has to be used to maintain sub-maps of features in each region.

The success of such an approach relies on the method used for inferring migration from one sub-map to another. This has been successfully done by treating each sub-map as a node of a graph by coalescing observations into independent local coordinates, and optimising the resulting graph globally (Eade and Drummond [2007]). However, still there is an upper limit to the number of landmarks that can be associated in pose estimation within a given node. It has been shown that the accuracy of the estimated pose can be effectively increased by increasing the number of landmarks used. Some other approaches show the possibility of pruning weak links of the inverse covariance matrix to achieve a sparse approximation that allows efficient and scalable SLAM (Thrun et al., Walter et al. [2007]). Later, sub-optimal solutions for SLAM have been suggested which exploit the probabilistic nature of the problem by assuming the camera motion to be known to make each observation independent (Eade and Drummond [2006], Montemerlo et al. [2002]). However, it has been shown that such approaches could become inconsistent with time (Bailey et al. [2006b]). Other approaches try to reduce the dimensionality by identifying higher order structures with a small dimensionality in the environment (Brunskill and Roy [2005]). Fusing information gathered from existing landmarks to create higher order structures can only reduce the dimensionality of the problem locally as the number of such structures also increases with time. The possibility of approximating high-dimensional discrete distributions to a tractable representation has been previously employed successfully to reduce the computational complexity for recognising places based on their appearance (Cummins and Newman [2008]).

4.3 Overview of the Kalman filter

The Kalman filter (Kalman [1960]) is the optimal solution to estimation and prediction tasks in which the measurement and process noise are Gaussian and the system dynamics are linear. It is used to estimate the underlying state of a system under noisy measurements. In the literature there are different formulations of the

filter, and the most widely known derivation uses the state space model including the original paper it-self. Although the full derivation is beyond the scope of this work, we give the basic form of the final result for clarity.

Consider any dynamic system, where $\{t, t-1, t-2, \ldots, 0\}$ are the time steps taken by the system and t is the current time step. Let $\mathbf{x}_t$ represent the state of the system and $\mathbf{z}_t$ the measurement, both at time t. Also let $\mathbf{u}$ and $\mathbf{v}$ be the process and measurement noise of the system with covariances Q and R respectively. Then the process model f which relates the state $\mathbf{x}_{t-1}$ at time $t-1$ with the state prediction $\mathbf{x}_{t|(t-1)}$ at time t, and the measurement model h which relates the state prediction $\mathbf{x}_{t|(t-1)}$ with the measurement at time t of the filter respectively are:

$$\begin{aligned} \mathbf{x}_{t|(t-1)} &= f\left(\mathbf{x}_{(t-1)}\right) + \mathbf{u} \\ \mathbf{z}_t &= h\left[\mathbf{x}_{t|(t-1)}\right] + \mathbf{v} \end{aligned} \tag{4.1}$$

The extended Kalman filter linearises these models around the current state. If the process and measurement Jocobians are F and H respectively, the standard derivation shows that if the state covariance is Σ, the predicted state $x_{t|(t-1)}$ and the predicted covariance $\Sigma_{t|(t-1)}$ of the system have to be updated using the Kalman gain:

$$K_t = \Sigma_{t|(t-1)} H^T \left(H \Sigma_{t|(t-1)} H^T + R\right)^{-1} \tag{4.2}$$

In the alternate formulation, the objective of the Kalman filter is to minimise the mean squared error between the actual and estimated data. Thus, it provides the best estimate of the data in the mean squared error sense. The Kalman filter is commonly known as a recursive least squares (RLS) filter, and the alternate derivation draws similarities to the chi-square merit function. With Gaussian measurement noise, the chi-square merit function of the Kalman filter can be

written as:

$$\chi^2 = \sum_{i=1}^{n} \left[\frac{z_i - h\left(\tilde{x}_i\right)}{\sigma_i} \right]^2 \tag{4.3}$$

Representing the chi-square in the vector form and using the notation of the Kalman derivation, this can be written as:

$$\chi^2 = \left[\mathbf{z}_t - h\left(\mathbf{x}_{t|(t-1)}\right)\right] R^{-1} \left[\mathbf{z}_t - h\left(\mathbf{x}_{t|(t-1)}\right)\right]^{\mathrm{T}} \tag{4.4}$$

Minimising the chi-square merit function leads to the Kalman gain such that:

$$K_t = \left(\Sigma_{t|(t-1)}^{-1} + H^{\mathrm{T}} R^{-1} H\right)^{-1} H^{\mathrm{T}} R^{-1} \tag{4.5}$$

It can be shown that these two forms of the Kalman gain in Equation 4.2 and Equation 4.8 are mathematically equivalent.

Equation 4.2 gives the standard form of the Kalman gain. By inserting $\Sigma_t \Sigma_t^{-1}$ and RR^{-1} into that we obtain:

$$\begin{aligned}
K_t &= \Sigma_t \Sigma_t^{-1} \Sigma_{t|(t-1)} H^{\mathrm{T}} R^{-1} R \left(H \Sigma_{t|(t-1)} H^{\mathrm{T}} + R\right)^{-1} \\
&= \Sigma_t \Sigma_t^{-1} \Sigma_{t|(t-1)} H^{\mathrm{T}} R^{-1} \left(H \Sigma_{t|(t-1)} H^{\mathrm{T}} R^{-1} + I\right)^{-1} \\
&= \Sigma_t \left(I + H^{\mathrm{T}} R^{-1} H \Sigma_{t|(t-1)}\right) H \mathrm{T} R^{-1} \left(H \Sigma_{t|(t-1)} H^{\mathrm{T}} R^{-1} + I\right)^{-1} \\
&= \Sigma_t H^{\mathrm{T}} R^{-1} \left(I + H^{\mathrm{T}} R^{-1} H \Sigma_{t|(t-1)}\right) \left(I + H \Sigma_{t|(t-1)} H^{\mathrm{T}} R^{-1}\right)^{-1} \\
&= \Sigma_t H^{\mathrm{T}} R^{-1}
\end{aligned} \tag{4.6}$$

The information form of the covariance update for the filter is given by:

$$\Sigma_t^{-1} = \Sigma_{t|(t-1)}^{-1} + H R^{-1} H^{\mathrm{T}} \tag{4.7}$$

By substituting the inverse of the information form of the covariance update in Equation 4.7 into Equation 4.8 we obtain:

$$K_t = \left(\Sigma_{t|(t-1)}^{-1} + H^{\mathrm{T}} R^{-1} H\right)^{-1} H^{\mathrm{T}} R^{-1} \tag{4.8}$$

This is the same as the gain calculated from the chi-square equations, confirming that the gains are indeed equivalent.

Although these two forms are mathematically equivalent, we note that there is an important difference between them with respect to the space which they are operating on, which is the key insight which leads to our reduced dimensionality Kalman filter. Let us consider the part in the Kalman gain that has to be inverted. In both forms, the inverted part is a form of an information matrix as it inverts the sum of two covariance matrices. In Equation 4.2 the information matrix $(H\Sigma_{t|(t-1)}H^{\mathrm{T}} + R)^{-1}$ works in the space of the measurement vector, by projecting the system covariance $\Sigma_{t|(t-1)}$ through the measurement Jacobian H as $H\Sigma_{t|(t-1)}H^{\mathrm{T}}$ onto the measurement space. In contrast, the information matrix $(\Sigma_{t|(t-1)}^{-1} + H^{\mathrm{T}}R^{-1}H)^{-1}$ in Equation 4.8 works in the space of the state vector of the filter by first projecting the measurement covariance R through the Jacobian transpose H^{T} as $H^{\mathrm{T}}R^{-1}H$ onto the space of the state vector.

Although mathematically equivalent, these two representations can lead to different complexities when matrices are inverted. Let us assume that there are n filter dimensions and m observations at a given time. Equation 4.2 requires inverting a $m \times m$ matrix while Equation 4.8 requires inverting a $n \times n$ matrix. The Kalman filter is usually updated in the space of the measurements, which leads to the gain Equation 4.2. If there are more state variables than the number of measurements at a given time such that $n > m$, using Equation 4.2 will be more efficient. However, if the state vector of the filter is sufficiently small at a given time, updating in the space of the state variables using Equation 4.8 is more efficient. In practice, the state dimensionality is comparatively large. If the size of the state vector can be kept small enough to make $n < m$, Equation 4.8 can be used to update the filter efficiently irrespectively of the number of measurements.

In the next section we describe a method that can formulate an alternate state vector which has a smaller dimensionality compared to the original. To do so, we assume the special class of Kalman filters where the process noise is zero, so the state can be represented as:

$$\begin{aligned} \mathbf{x}_{t|(t-1)} &= \mathbf{x}_{t-1} \\ \mathbf{z}_t &= h\left[\mathbf{x}_{t|(t-1)}\right] + \mathbf{v} \end{aligned} \tag{4.9}$$

The goal is to reduce the dimensionality of the state vector of the system to make its size smaller compared to the measurement vector.

4.4 Reduced Dimensionality Kalman Filter

If we limit our attention to a filter with zero process noise where the state can assumed to be static, it is possible to represent the mean of this distribution as a point in $n-D$ space with its uncertainty given by the covariance matrix as a hyper ellipse. Measurement information continuously reduces this uncertainty along some directions. With time, gradually the uncertainty along many directions becomes comparatively small, making further information along those directions redundant or less useful. Hence, if the filter starts at $t = 0$ with a totally unknown state and continues gathering information until $t = t_k$, the mean state of the filter, $\mathbf{x}_{t_k}$ can be frozen and future changes to the mean can be represented as offsets from this mean. This enables decomposing the covariance matrix into reasonably certain and mostly uncertain dimensions by singular value decomposition. Let the covariance of the EKF be Σ_{t_k} at $t = t_k$. Then:

$$\Sigma_{t_k} = UDU^T \tag{4.10}$$

where, U is the set of singular vectors of Σ_{t_k} and D is the diagonal matrix of singular values. The singular vectors represent de-correlated uncertainty directions with variances proportional to their singular values. We partition D as D_s, the significant (large) set and D_i the insignificant (small) set.

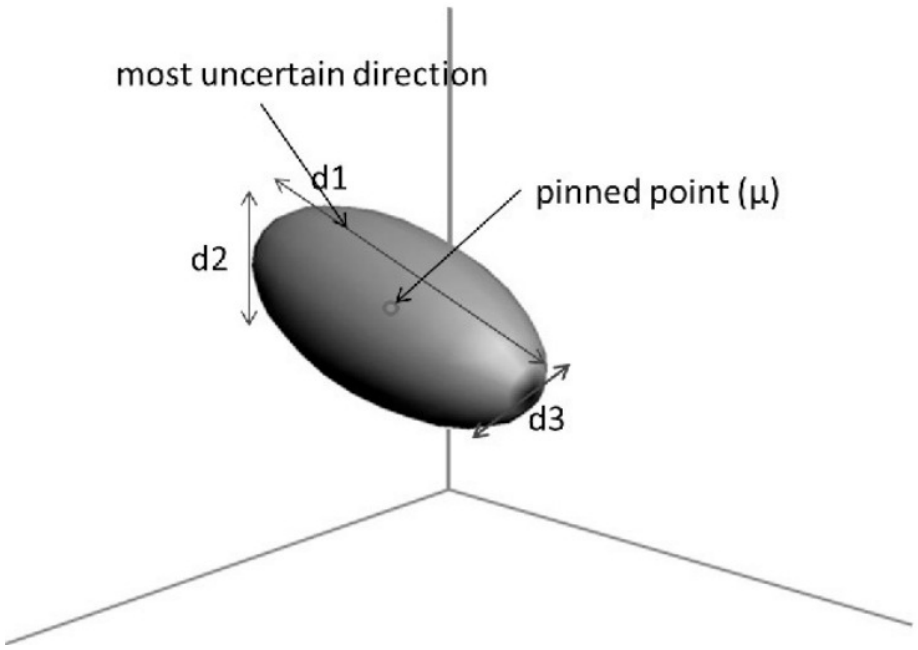


Figure 4.1: An uncertainty ellipsoid in $3-D$ around a pinned point $\mathbf{x}$, where $d1$ represents the most uncertain direction.

U is also partitioned as U_s, vectors corresponding to D_s and U_i, vectors corresponding to D_i. Σ_t can now be written in block form:

$$\Sigma_{t_k} = \left[\begin{array}{c|c} U_s & U_i \end{array}\right] \left[\begin{array}{c|c} D_s & 0 \\ \hline 0 & D_i \end{array}\right] \left[\begin{array}{c} U_s^T \\ \hline U_i^T \end{array}\right] \tag{4.11}$$

Therefore, the original covariance matrix becomes:

$$\Sigma_{t_k} = U_s D_s U_s^T + U_i D_i U_i^T \tag{4.12}$$

As the second term of the above expression is comparatively small, the column space of U_s can be used as the reduced space ignoring small singular vectors. Let the significant vectors U_s extracted at $t = t_k$ be U_{st_k} and x'_t be the reduced state which is Gaussian distributed in the reduced space with a mean $\mathbf{x}'_t$ at $t(\geq t_k)$. The reduced mean $\mathbf{x}'_t$ relates to the original state mean $\mathbf{x}_t$ as:

$$\mathbf{x}_t = \mathbf{x}_{t_k} + U_{st_k}\mathbf{x}'_t \tag{4.13}$$

New state x'_t represents the variations of the original state around the point $\mathbf{x}_{t_k}$ along the directions of the column space of U_{st_k}. Initially we start with $\mathbf{x}_{t_k}$ being a zero vector, indicating our knowledge about uncertainty is zero along corresponding singular vectors in U_{st_k}. Subsequent observations can be projected onto the derived reduced space to gather information about the new state x'_t. The projected covariance matrix Σ_{t_k} is obtained as:

$$\Sigma'_{t_k} = U_{st_k}^T \left(\Sigma_{t_k} - U_i D_i U_i^T\right) U_{st_k} \tag{4.14}$$

The dimensionality of x'_t can be kept quite small, compared to the original state x_t. For all time steps $t(\geq t_k)$, information can be collected to update x'_t by changing its mean and the covariance. This makes the reduced state time-dependent. To obtain the prediction equation in the reduced space, the linearised EKF states at time steps t and $t-1$ can be decomposed according to Equation 4.13. As we are assuming zeros process noise, the process model becomes the

identity. Substituting the decomposed states into the process equation yields:

$$\mathbf{x}_{t_k} + U_{st_k}\mathbf{x}'_t = \left(\mathbf{x}_{t_k} + U_{st_k}\mathbf{x}'_{t-1}\right) \tag{4.15}$$

making the predicted state same as the previous state:

$$\mathbf{x}'_t = \mathbf{x}'_{t-1} \tag{4.16}$$

Similarly, if the measurement is z_t with the model Jacobian H_t and measurement noise v_t, the update equation, after substitution becomes:

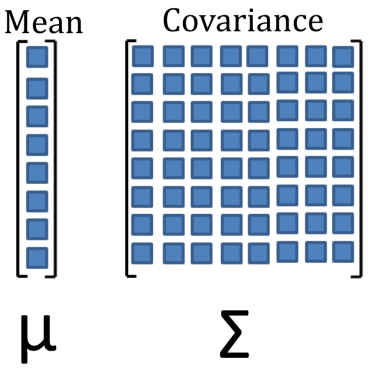
$$z_t = H_t\left(\mathbf{x}_{t_k} + U_{st_k}\mathbf{x}'_t\right) + v_t \tag{4.17}$$

which can be modified as:

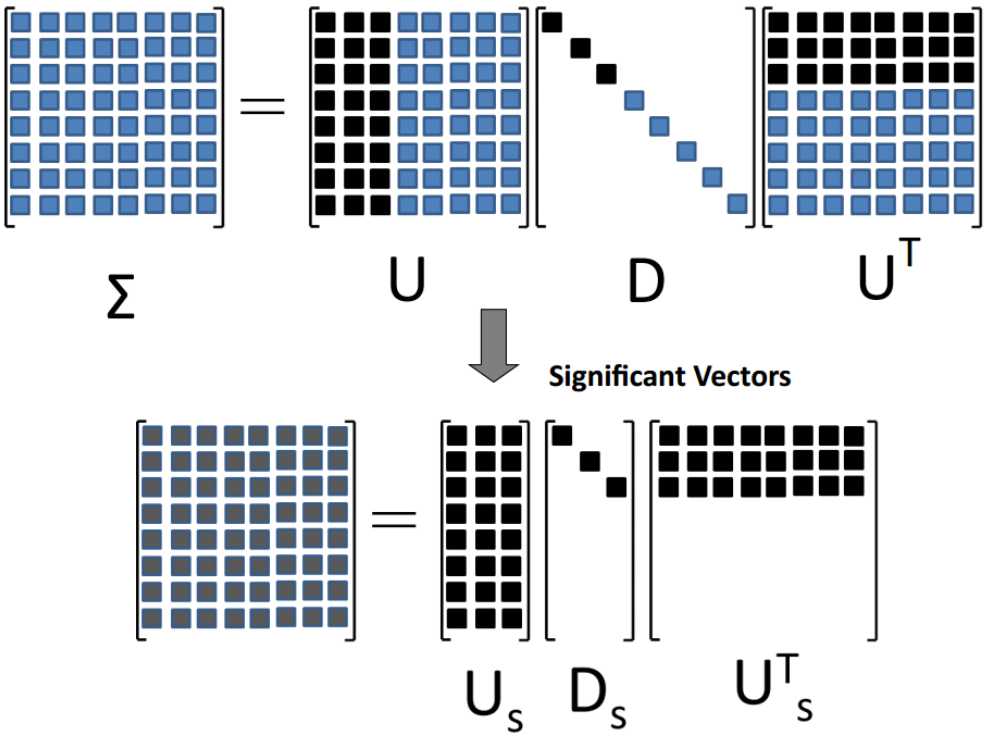
$$z_t - H_t\mathbf{x}_{t_k} = H_t U_{st_k}\mathbf{x}'_t + v_t \tag{4.18}$$

Here $z_t - H_t\mathbf{x}_{t_k}$ becomes the modified observation and $H_t U_{st_k}$ the projected Jacobian.

This decomposition can be better explained using a pictorial representation. As shown in Figure 4.4, the filter covariance matrix can be represented using a few of its significant singular values by compromising the accuracy a little.

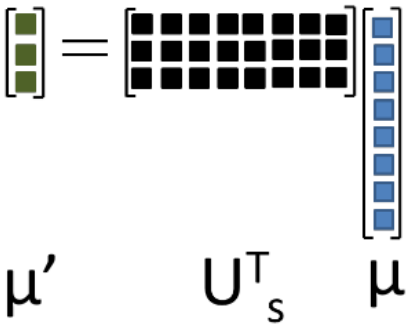


(a) Mean and the covariance of original the Kalman filter

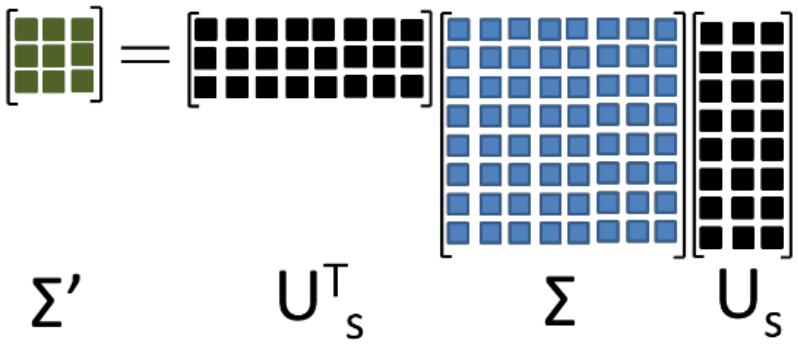


(b) Decomposition using SVD

Using the reduced space extracted from the covariance matrix, the original mean vector and the covariance matrix can be reduced as shown in Figure 4.4.



(c) Reduced mean



(d) Decomposition using SVD

In a Kalman filtering set-up, this reduced space representation can be used to efficiently fuse new information obtained through continuous measurements.

The dimensionality reduction described so far works only when the state is static. As the camera pose keeps changing it is necessary to account for these changes through a different approach as described later.

4.4.1 Adding New Landmarks

The computational saving achieved by reducing the dimensionality of the filter can be spent by admitting more variables into the filter to be measured (thus increasing its dimensionality again). A new variable l, can be directly added to the reduced state to obtain an augmented state. This requires increasing the size of $\mathbf{x}'_t$ and appending rows and columns to U_{st} and Σ'_t. The augmented state estimate $\hat{\mathbf{x}}'_t$ and reduced vector space $\hat{U}_{st_k}$ now can be modified as:

$$\hat{\mathbf{x}}'_t = \begin{pmatrix} \mathbf{x}'_t \\ l \end{pmatrix} \qquad \hat{U}_{st_k} = \begin{pmatrix} U_{st_k} & 0 \\ 0 & I \end{pmatrix} \tag{4.19}$$

with the covariance:

$$\hat{\Sigma}'_t = \begin{pmatrix} \Sigma'_t & 0 \\ 0 & \sigma \end{pmatrix} \tag{4.20}$$

where σ is the initialising uncertainty. It has to be noted that this augmentation also increases the dimensionality of the original space by the same number of dimensions. Therefore, the frozen mean $\mathbf{x}_{t_k}$ has to be augmented with zeros to obtain a modified mean $\hat{\mathbf{x}}_{t_k}$. The relationship between the state estimate in the original space and the reduced space becomes:

$$\hat{\mathbf{x}}_t = \begin{pmatrix} \mathbf{x}_t \\ l \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{t_k} \\ 0 \end{pmatrix} + \begin{pmatrix} U_{st_k} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \mathbf{x}'_t \\ l \end{pmatrix} \tag{4.21}$$

Here we have used the "hat" to denote augmented vectors and matrices. For clarity we can simply remove the "hat" and use these variables to replace their un-augmented counterparts.

4.4.2 Iterative State Reduction

With continuous augmentations the reduced state $x'_{k|k}$ will also start growing. To keep the dimensionality of $x'_{k|k}$ manageable, we decompose it continuously by distributing newly learned knowledge over the frozen mean $\mathbf{x}_{t_k}$ and select a new reduced basis. Let us decompose Σ'_t as shown in Equation 4.32 at a time step $t = t_n (\geq t_k)$ and denote the new basis by U'_{st_n}. Let x''_t be the new state, Gaussian distributed with a mean $\mathbf{x}''_t$. Thus $\mathbf{x}'_t$ for any $t (\geq t_n)$:

$$\mathbf{x}'_t = \mathbf{x}'_{t_n} + U'_{st_n}\mathbf{x}''_t \tag{4.22}$$

By substituting this into Equation 4.31 we can have:

$$\mathbf{x}_t = \mathbf{x}_{t_k} + U_{st_k}\left(\mathbf{x}'_{t_n} + U'_{st_n}\mathbf{x}''_t\right) \tag{4.23}$$

The above equation can be simplified as:

$$\mathbf{x}_t = \mathbf{x}_{t_k} + U_{st_k}\mathbf{x}'_{t_n} + U_{st_k}U'_{st_n}\mathbf{x}''_t \tag{4.24}$$

The reduction in Equation 4.24, distributes the information learnt up to t_n over $\mathbf{x}_{t_k}$ to obtain an updated mean $\mathbf{x}_{t_n}$ such that:

$$\mathbf{x}_{t_n} = \mathbf{x}_{t_k} + U_{st_k}\mathbf{x}'_{t_n} \tag{4.25}$$

With this the original reduced basis gets rotated towards the new basis to yield a new reduced basis U_{st_n} as:

$$U_{st_n} = U_{st_k}U'_{st_n} \tag{4.26}$$

This distribution can be done on a frame-to-frame basis after each update, which makes the reduced basis U_{st_n} and the frozen mean $\mathbf{x}_{t_n}$ time-dependent.

In practice, the significant set of vectors will remain around 20, as all other singular values are very small or zero. Irrespective of the size of the original state, this makes it possible to keep the reduced space dimensionality bounded.

4.5 Dimensionality Reduction in SLAM

Camera state and new features have to be handled separately when reducing the dimensionality in SLAM. A state vector x, composed of the camera and all the map features, can be written as:

$$\mathbf{x} = \left(x_c^T, y_1^T, y_2^T, ..., y_n^T\right)^T \tag{4.27}$$

where a scene $3-D$ point i is represented using a vector y_i and the camera by x_c.

4.5.1 Reduction with the Camera

To model the desired variation while retaining the ability to do standard operations on the camera, here we perform reduction only upon landmark states. As the first step, landmarks observed up to time $t = t_k$ are decomposed, keeping the camera state intact by directly transferring it onto the reduced space. Transfer stacks the camera on top of the reduced state, which initially represented only the variation of landmarks.

Let us represent the state at time t_k, which is composed of camera parameters $x_{c_{t_k}}$ and the landmarks y_{t_k} as a Gaussian:

$$\mathbf{x}_{t_k} = \begin{pmatrix} \mathbf{x}_{c_{t_k}} \\ \mathbf{x}_{y_{t_k}} \end{pmatrix} \qquad \Sigma_{t_k} = \begin{pmatrix} \Sigma_{c_{t_k}} & \Sigma_{c_{t_k} y_{t_k}} \\ \Sigma_{y_{t_k} c_{t_k}} & \Sigma_{y_{t_k}} \end{pmatrix} \tag{4.28}$$

For dimensionality reduction, first the covariance matrix of landmarks $\Sigma_{y_{t_k}}$, can be decomposed according to Equation 4.11 and then significant dimensions $U_{sy_{t_k}}$ can be extracted as the reduced space. As described earlier, we fix the current state of the landmarks $\mathbf{x}_{y_{t_k}}$. In addition to the steps described in the previous section, to directly transfer the camera, $x'_{y_{t_k}}$ onto the reduced space we have to extend its dimensions. Therefore, the reduced vector space $U_{sy_{t_k}}$ has to be extended by adding a set of orthogonal axes. Let the augmented reduced state be x'_{t_k} and the vector space be U_{st_k}. Then:

$$\mathbf{x}'_{t_k} = \begin{pmatrix} \mathbf{x}_{c_{t_k}} \\ \mathbf{x}'_{y_{t_k}} \end{pmatrix} \qquad U_{st_k} = \begin{pmatrix} I & 0 \\ 0 & U_{sy_{t_k}} \end{pmatrix} \tag{4.29}$$

The fixed point also has to be augmented with a set of zeros as we are transferring camera parameters onto the reduced space. The modified fixed point then becomes:

$$\mathbf{x}_{t_k} = \begin{pmatrix} 0 \\ \mathbf{x}_{y_{t_k}} \end{pmatrix} \tag{4.30}$$

The original state x_t relates to the reduced state at $t \geq t1$ through:

$$\mathbf{x}_t = \begin{pmatrix} 0 \\ \mathbf{x}_{y_{t_k}} \end{pmatrix} + \begin{pmatrix} I & 0 \\ 0 & U_{sy_{t_k}} \end{pmatrix} \begin{pmatrix} \mathbf{x}_{ct} \\ \mathbf{x}'_{yt} \end{pmatrix} \tag{4.31}$$

The covariance in the reduced space can be easily obtained by projecting the original covariance onto the reduced space U_{st_k}:

$$\Sigma'_{t_k} = U_{st_k}^T \left(P_k^W - \begin{pmatrix} 0 & 0 \\ 0 & U_{i_y} D_{i_y} U_{i_y}^T \end{pmatrix} \right) U_{st_k} \tag{4.32}$$

Camera dynamics can be incorporated by working on this space where we have augmented the full camera state instead of its variations.

4.5.2 Prediction in the Reduced Space

When landmarks are static, the prediction involves estimating the camera and its covariance with cross-covariances (Civera et al. [2009]). As we transfer the camera directly onto the reduced state x'_t, camera parameters can be predicted in the usual manner while keeping x'_{y_t} unchanged to obtain the predicted state $x'_{t|(t-1)}$. The process noise corresponding to the camera has to be added to the camera covariance block in the reduced covariance Σ'_t to obtain the predicted covariance $\Sigma'_{t|(t-1)}$.

4.5.3 Measurements in the Reduced Space

State updating is done by projecting the measurement model onto the reduced space. Measurement model Jacobian H_t, can be projected onto reduced space by a right multiplication. For clarity, if we drop the subscript t and denote the

Jacobian as H:

$$H_s = HU_{st_k} \tag{4.33}$$

The innovation covariance can be written as:

$$S_s = H_s \Sigma_{t|(t-1)} H_s^T + I \tag{4.34}$$

Therefore, the reduced Kalman gain becomes:

$$K_s = x'_{t|(t-1)} H_s^T S_s^{-1} \tag{4.35}$$

If the predicted measurement is h, the reduced state can be updated as:

$$x_t = x_{t|t-1} + K_s(z - h) \tag{4.36}$$

4.6 Complexity Analysis

When all landmarks are static, the prediction simply becomes predicting camera parameters according to a dynamic model. As we directly transfer the camera onto the reduced space, the prediction complexity remains the same. Dimensionality of the camera covariance remains the same as it is not reduced. The covariance of landmarks stays intact as landmarks are static. The matrix block that represents the cross-covariances between landmarks and camera needs to be multiplied by the camera Jacobian. Let the dimensionality of the camera and landmarks to be n_c and n_l respectively. Let n_d be the dimensionality of the reduced space. In the reduced space, the complexity of this multiplication becomes $\mathcal{O}(n_c^2 n_d)$ compared to the $\mathcal{O}(n_c^2 n_l)$ complexity in the original space.

Updating the state requires inverting the innovation covariance of the filter. Let n_L and n_D be the dimensionality of full states including the camera in the original space and in the reduced space, respectively. The complexity of updating with a single landmark becomes $\mathcal{O}(n_L^2)$ in the original space. To obtain the Kalman gain, the innovation covariance has to be inverted. The complexity of inverting an n dimensional matrix is $\mathcal{O}(n^3)$. However, with the sequential innovation Kalman filter formulation (Evensen [2003]), if the number of current measurements is limited to some number n_m which is smaller than the state dimensionality, the overall complexity reduces to $\mathcal{O}(n_m n^2)$. In SLAM, the number of observable features n_m at a given time is smaller compared the total number of landmarks. This reduces the complexity of the inversion from $\mathcal{O}(n^3)$ to $\mathcal{O}(n_m n^2)$. When information is projected onto the reduced space, that information is distributed over all reduced states, making a sequential approach impossible. Therefore, complexity has to remain cubic in the reduced space. However, dimensionality of the reduced space is extremely small compared to the original space, where the complexity is $\mathcal{O}(n_D^3)$. Most importantly, the proposed method keeps n_D around 20 irrespective of the original state dimensionality n_L.

4.7 Experiments

In order to prove the validity and accuracy of the method proposed, first we used synthetically generated data. In the experimental set-up we generated a set of $3-D$ landmarks along with a synthetic camera sequence. All landmarks were inverse-depth coded. At each time step t, we projected the $3-D$ landmarks onto the corresponding camera plane C_t. Each projected feature was added with random Gaussian noise with a unit pixel standard deviation. To make the comparison a fair one, we used the same set of generated noisy data points to evaluate the accuracy of both scenarios. These projected noisy feature points with their data associations in the previous image were used to estimate camera parameters in an EKF set-up. To evaluate the reduced dimensional EKF, we shifted from the

standard filter to the reduced filter after learning the environment for some time. After each estimation step, inverse depth was extracted and normalised by the mean inverse depth to remove scaling ambiguity. Synthetic ground truth data were also normalised in a similar manner.

When calculating the average inverse depth from estimated landmarks, to obtain a meaningful average depth, selected features need to be sufficiently certain. In other words, if there are newly initialised landmarks with highly uncertain depth estimates, the average will become erroneous. If the estimated depth for a landmark is ρ such that $\rho > 0$ with a covariance Σ_ρ to avoid such situations, we set the selection criteria of the landmark to be:

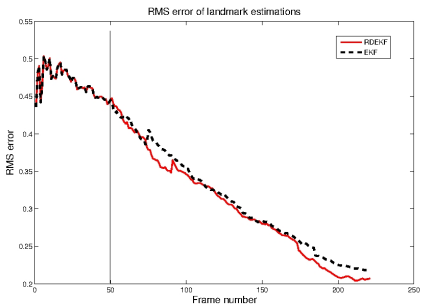
$$\rho - \sqrt{\Sigma_\rho} > 0 \tag{4.37}$$

In our experiment we took the difference between the estimated inverse depth ρ_e and the ground truth ρ_g for each selected landmark after normalization. Then we divided this difference by the inverse depth covariance of the estimated landmark in order to give a lower weight to landmarks with higher uncertainty. To remove the error induced by the scale of the covariance (due to multiplication), the result has to be multiplied by the average inverse depth again. Let $\mathbf{x}_e$ and $\mathbf{x}_g$ be the averages for estimated and ground truth inverse depths respectively. If n is the number of landmarks, the final RMS error for each frame can be obtained as:

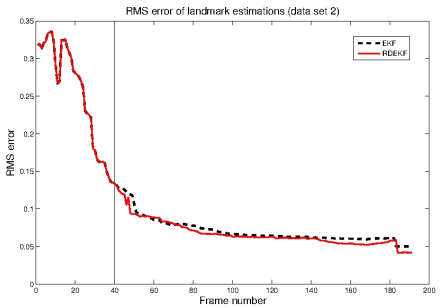
$$E_{RMS} = \sqrt{\sum \left[\left(\frac{\rho_e}{\mathbf{x}_e} - \frac{\rho_g}{\mathbf{x}_g} \right) \frac{\mathbf{x}_e}{\sqrt{\Sigma_\rho}} \right]^2 \div n} \tag{4.38}$$

Figure 4.2 shows the RMS error of estimated inverse depths against the frame number. The vertical line indicates the instance we started reducing the dimensionality of the system. Until the environment is learnt sufficiently for reduction,

we have to rely on the standard EKF. The dotted line graph shows the error of the EKF-based landmark estimation and the full-line graph shows the error of the reduced dimensional EKF (RDEKF) based landmark estimation. The graphs clearly indicate that the proposed RDEKF yields similar or better results compared to the standard EKF. Other than the dimensionality reduction, selecting dominant modes bundles the movement of individual landmarks along the global trend of the system. The error reduction can be attributed to the outlier tolerance we gain through dimensionality reduction.



(e) Dataset 1



(f) Dataset 2

Figure 4.2: RMS error of landmark estimations with conventional EKF and proposed reduced dimensionality EKF.

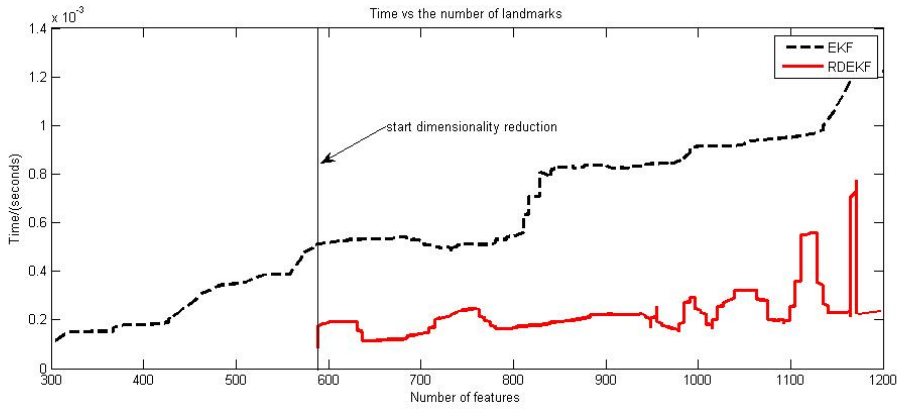


Figure 4.3: Time complexity of the update step for a real data sequence

.

Figure 4.7 shows the time taken by the update step of the Kalman filter for real data. Here we have considered only the update step, as predictions are almost the same for both methods. In the figure, the values have been slightly median-filtered to remove sudden spikes that occur in measuring the time complexity. The graph clearly shows that we can keep the execution time of the RDEKF almost linear.

4.8 Discussion

In this chapter we introduced a dimensionality reduction technique to handle the complexity growth of the extended Kalman filter. Although the Kalman filter is no longer the state-of-the-art with current sparse matrix methods, we believe the Kalman filter has still considerable potential, especially for collaborative SLAM.

The PTAM has later been extended to accommodate multiple trackers to map the environment through a sub-mapping approach (Robert et al. [2011]). It would be very efficient if it is possible to replace the back end of such a system with a new sparse matrix method. However, such a back-end will impose some other limitations in a multi-camera set-up. When multiple cameras are updating the same sub-map, it is impossible to do a bundle adjustment within that sub-map in an incremental fashion, making methods like incremental smoothing and mapping (ISAM) (Kaess et al. [2012]) less appropriate, as they cannot have multiple roots. The possible solution of merging maps together requires marginalising all camera poses from each map to build a secondary map only with landmarks before fusing them together (Cunningham et al. [2010]). This marginalization becomes tedious when the graphs are large.

We propose a method of representing SLAM as a graph of coalesced observations (Eade and Drummond [2007]) as a solution. Such a graph can be easily extended to accommodate multiple cameras, each coalescing its information

into a selected node, and global consistency can be maintained separately. This method represents observations relative to an active node and has some similarities with relative bundle adjustment (Sibley et al. [2010]) where landmarks are represented relative to their observed coordinate system. However, as the camera is marginalised from each node when coalescing observations, multiple camera information can be fused independently into each node. Compared to existing methods which require camera marginalization from a large bundle-adjusted graph (Cunningham et al. [2010]), fusing information into a node will be much more efficient. However, fusing new information should be done only along uncertain directions. Our dimensionality reduction technique can be used to identify such directions, which in turn increases the number of landmarks that can be handled by each node for improved accuracy (Strasdat et al. [2010a]).

When extending the method we propose in this chapter to large scales, the solution drifts away as the linearization errors are permanently baked into the system as the system marginalises old camera poses. In addition, here we have assumed a camera-centered coordinate system. In the next chapter we show how our method can be extended to large scales by retaining all camera poses in the filter. We further extend the method using the relative formulation of landmarks to make it more linear inspired by relative bundle adjustment (RBA) for SLAM.

Chapter 5

Reduced Dimensionality Extended Kalman Filter for SLAM in a Relative Formulation

In vision-based SLAM, the number of features used plays a central role on the accuracy of the final pose estimate, as argued in the previous chapter. Increasing the number of features increases the accuracy of the final pose estimation, but increases the dimensionality of the underlying optimization problem.

In this chapter we analyse the effect of the number of features incorporated into the system on the final estimation uncertainty. We show that the actual number of dimensions that has to be optimised is far less than the number of variables in the problem, extending the work in previous chapter. This is true, as the structure of the problem is entirely determined by the camera motion rather than the number features encompassed. The number of features used only determines its noise tolerance.

Furthermore, we extend this notion to build a consistent real-time SLAM system for medium-scale environments, and the computational complexity of the filter is kept manageable by reducing the dimensionality as information is acquired.

By fusing information only along the most uncertain directions of the filter, we show how filter complexity can be reduced dramatically.

In this chapter, we represent landmarks in a relative formulation as being different from the method introduced earlier. We also formulate the problem by including all camera poses, without marginalising them. This makes the problem more linear, yielding more accurate results, enabling extension of the dimensionality reduction to medium-scale problems.

Using experimental results, finally we prove that our method can handle more information and yield more accurate estimates compared to the state-of-the-art methods.

5.1 Introduction

In a SLAM system, with time, the number of measurements of landmarks in the environment increases rapidly. This becomes problematic for SLAM systems which work in real time, imposing an upper limit on the maximum number of landmarks that can be handled by the system. In contrast, it is the amount of information acquired that improves the accuracy of the final relative pose estimation of the system. Hence, reducing the number of features increases the speed, compromising estimation accuracy.

An example of the inconsistency problem explained in the previous chapter arises when the uncertainty covariance for each landmark has an elongated shape

(This is a frequent occurrence because the distance of a landmark from the camera has much greater uncertainty than the position of the landmark in the field of view). When the cameras orientation estimate drifts from the truth, this covariance structure points in the wrong direction. If the orientation error is corrected by a loop closure, then the uncertainty ellipse no longer correctly represents the state of knowledge about the landmarks position, and subsequent observations of the landmark can lead to false reductions in uncertainty that place the landmark in the wrong position.

Newer methods like PTAM (Klein and Murray [2007]), ISAM (Kaess et al. [2008]), DWO (Strasdat et al. [2011]), and coalesced observations SLAM (Eade and Drummond [2007]) solve this problem by retaining old views and performing bundle adjustment each time a key frame is added to the view set.

These methods take advantage of the sparsity of the full problem. PTAM uses the Schur complement, which depends on the fact that each observation couples a landmark with a camera and there are no observations coupling two landmarks directly. This reduces the complexity to the cube of the number of key-frames in the bundle adjustment stage. ISAM further exploits the sparsity that arises because each landmark is only seen by a (possibly small) subset of the views in a large SLAM problem. It further orders the variable elimination so that if future frames that only observe recently observed landmarks, the majority of calculations can be reused without having to be repeated.

These methods are typically restricted to sparse sampling in time of key-frames, because the complexity depends on the number of frames in the bundle adjustment. Key-points are tracked densely in time to minimise the occurrence of false correspondences and to provide a pose initialisation of the new key-frame for bundle adjustment. Coalesced observations SLAM uses dense sampling in time by operating a Kalman filter within the (approximately) linear region around each key-frame. The range of this linear region is made larger by the use of inverse depth coordinates for landmarks relative to each key-frame rather than Cartesian coordinates (either locally or globally). The use of key-frame-relative coordinates

is possible because the poses of key-frames are retained as optimization parameters.

In the previous chapter we took an alternative approach to reducing the time complexity of the Kalman filter by observing that the energy spectrum of the covariance matrix is typically concentrated into a much smaller number of dimensions than are present in the filter. This means that the covariance matrix can be closely approximated by a matrix of reduced rank - i.e. that the SLAM problem can be parametrised in terms of a small set of dimensions that represent the degrees of freedom with significant uncertainty.

In this work, we take this approach further and apply it to the full estimation problem containing all camera poses as well as landmark positions. This approach has a number of advantages:

- It makes it possible to operate a filter with many thousands of state variables using a re-parametrization with only a few tens (e.g. 20) of dimensions.
- It enables us to optimise using *every* frame of the video feed. Further, we show that it is advantageous to do this because the resulting dimensionality of the filter is reduced, despite the increase in dimensionality of the underlying state space due to the inclusion of the pose of every camera. Similarly, we show that the approach benefits from observing a *greater* number of landmarks in each frame because again the increased information input results in a *lower* number of reduced dimensions.
- It allows us to use a camera-relative representation for landmarks, as used byt Sibley et al. [2010] and Eade and Drummond [2007]. This greatly relieves the optimization from its inherent non-linearity limitation.
- It produces a more accurate result compared to some state-of-the-art methods.

This chapter is organised as follows. For clarity, we first repeat some of

the derivation in the previous chapter to show how to parametrise the filter in terms of uncertainty and work with a reduced number of uncertain dimensions, maintaining a covariance matrix only in this reduced space (see Section 5.2). Then in Section 5.2.1, we show how to decompose the state of the full SLAM problem to obtain a reduced system. Section 5.3 describes how to represent the relative kinematic-chain in a filter. Then we describe how landmarks are initialised efficiently using the sequential nature of the filter-based approach for SLAM (see Section 5.4). The landmark initialization scheme can be thought of as a bundle adjustment as we are adding new landmarks to the filter with all its past observations (see Section 5.5). In the experimental results section (5.6) we compare the translation accuracy of our system with the state-of-the-art double window optimization for SLAM. Then we discuss the limitations of the proposed method (see Section 5.7). Finally we conclude with a summary of the content (see Section 5.8).

5.2 Reducing the Kalman Filter Dimensions

In SLAM, the structure of the system is governed by the camera motion. Knowing the camera motion is equivalent to knowing the structure of the environment, as $3-D$ landmark locations can be readily recovered using that information. In the previous chapter it has been shown that the filter state $\mathbf{x}_t$ can be reduced to a small number of parameters. In this chapter, we give some details of the method for completeness and clarity.

Let the covariance of the EKF be Σ_{t_k} at some time instant. Then, as shown in the previous chapter, it can be decomposed as:

$$\Sigma_{t_k} = U_{t_k} D_{t_k} U_{t_k}^{\mathrm{T}} \tag{5.1}$$

where U_{t_k} is the set of singular vectors of Σ_{t_k} and D_{t_k} is the diagonal matrix of the corresponding singular values. The singular vectors represent de-correlated uncertainty directions with variances proportional to their singular values. We partition D_{t_k} into $D_{s_{(t_k)}}$, the significant set and $D_{i(t_k)}$ the insignificant set of the dimensions. U_{t_k} is also partitioned into $U_{s(t_k)}$, vectors corresponding to $D_{s(t_k)}$ and $U_{i(t_k)}$, vectors corresponding to $D_{i(t_k)}$.

Let x'_t be the reduced state which is Gaussian distributed at time $t_n(\geq t_k)$, as before.

The SVD decomposition done in the earlier chapter can work only when the process noise of the variables that are decomposed is zero. Earlier the pose at each time has been marginalised from the system to obtain the next pose at t, adhering to the conventional Kalman filtering approach for SLAM. Pose parameters at $t-1$ can be marginalised only after obtaining an estimate of the pose parameters at time t through the process model. As these pose parameters are subject to the process noise, they have to be treated differently from the landmarks which are not subject to process noise. However, it can be noted that, if all poses are retained in the system without marginalization, these parameters will remain static without being subjected to a process model. In this case, new poses have to be appended in the same way that the landmarks are appended to the system.

In this work we follow the second approach, where we retain all camera poses, as this removes the inherent non linearity problem of the Kalman filters.

5.2.1 State Decomposition

As is shown experimentally, the reduced dimensionality can be kept sufficiently small compared to the original state dimensionality, making above estimation efficient.

Different from the previous chapter, here we maintain all camera poses in the filter state, without any marginalization to reduce the non-linearity problem Strasdat et al. [2010a] of the conventional filter based solution. When the state contains the full trajectory of the camera, each camera pose is not subject to process noise. This is different from the previous formulation where the camera state has to be treated separately from $3-D$ scene points as the camera has to change dynamically as a consequence of the marginalization.

5.2.2 Iterative State Reduction

As has been shown in the previous chapter, with continuous exploration the reduced state $\mathbf{x}_t^{'}$ size will also start to grow as new variables are appended. Here, it will grow rapidly than before as we are including the whole camera trajectory in the system. To keep the dimensionality of $\mathbf{x}_t^{'}$ manageable, we decompose it continuously by distributing newly-learned knowledge over the previously fixed state $\mathbf{x}_{t_k}$ and selecting a new reduced basis. Let us decompose Σ_t' as shown in before, at a time step $t_n(\geq t_k)$ and denote the new basis by $U'_{s(t_n)}$. By following the same derivation as in the previous chapter, we can reduce this new state.

5.3 Relative Landmark Representation with the Full Trajectory

The relative parametrization of SLAM represents each $3-D$ scene point relative to its originating coordinate system (relative to the initialised frame), instead of a globally privileged coordinate system (Sibley et al. [2010]). This renders a better linearity to the problem as each camera pose can change by minimally affecting the landmarks initialised on the corresponding frame. Here we use the

same approach, but within a filtering paradigm.

With this parametrization, at a given time the state vector $\mathbf{x}_t$, composes each camera x_{c_i}, $i \in \{1, \ldots, n\}$ with 3D scene points $y_j^{c_i}$, $j \in \{1, \ldots, m\}$ initialised on each frame as given in Equation 5.2.

$$\mathbf{x}_t = \left(x_{c_1}, y_1^{c_1}, y_2^{c_1}, y_3^{c_1}, \ldots, x_{c_m}, y_j^{c_m}, y_{j+1}^{c_m}, y_{j+2}^{c_m}, \ldots\right)^{\mathrm{T}} \tag{5.2}$$

Here a $3-D$ scene point initialised on the camera c_i is defined by a $3-D$ vector $y_j^{c_i}$. For an incremental update on the camera c_k, the scene point $y_j^{c_i}$ has to be projected all the way through the kinematic chain from c_i to c_k.

5.3.1 Relative Representation within a Filter

In bundle adjustment as all camera poses under consideration are included in the optimization with corresponding measurements, a given camera pose can change consistently. Such a relative formulation will be useless in a filter as the final transformation can be obtained by changing any incremental pose of the kinematic chain to project a landmark onto the current frame. In other words while projecting a landmark from camera c_i to c_k, the correct transformation can be obtained by changing any camera pose c_j such that $i \leq j \leq k$ incrementally. Although, the uncertainty (covariance) of past camera poses remains small, which can keep these changes small. However, as there are no past observations as in relative BA Sibley et al. [2010] to anchor a particular incremental pose within the chain, it will slowly drift the intermediate estimates.

One possible solution would be to update the filter with past measurements. This will make it similar to a local bundle adjustment, but with a separately maintained covariance. In such an approach, using measurements which have been used already to update the filter could easily lead to an over-confident

covariance estimate. This can be mitigated by combining the system with a landmark initialization scheme.

Because of the high non-linearity which could drift the final pose estimate, it is quite common to initialise new landmarks with a separate information filter, and later add them into the optimiser (Eade and Drummond [2007], Sola et al. [2008]). This delayed feature initialization discards some information that can contribute to improving the filter estimate.

As shown in the Section 5.4, we initialise landmarks using the same delayed approach with separate information filters. Instead of updating the main filter only with the current measurement at the time of transfer, while transferring the newly initialised landmark from each individual information filter, we fuse all past measurements which were used to update the information filter as a batch when transferring them to the global optimiser. Newly added landmarks contribute to updating the underlying filter in a relative BA formulation incorporating all past measurements and camera poses. This approach is beneficial in three ways. First, it eliminates the information loss caused by delaying the landmark initialization. Second, it can anchor each camera pose, eliminating the relative update problem. Third, this enables us to control the available information per landmark. For instance, after a landmark initialization if there are not enough measurements, the landmark uncertainty will remain without reducing. With our approach we can control the number of available measurements to shrink the landmark uncertainty to a sufficiently small value without hindering the dimensionality reduction. Although this increases the size of the measurement vector used in the optimization, the dimensionality of the reduced state does not change.

5.4 Landmark Initialization

At the beginning the system's main Kalman filter is initialised with triangulated landmarks. Then on each frame, while updating the main filter, new landmarks are randomly initialised through independent information filters by setting the inverse depth to the average inverse depth of the previous frame. Then all newly initialised landmarks are updated given the camera pose estimates of the main Kalman filter (Eade and Drummond [2007]).

After accumulating sufficient parallax, landmarks are appended to the main filter and the filter is updated with all measurements of each newly added landmark from its initialization. This brings the uncertainty of the landmark to a small value while avoiding the same information being fused into the system. Landmarks are discarded without adding to the main filter if are insufficient measurements within a past number of frames. When updating the filter, we use past measurements of newly added landmarks, making our approach locally similar to bundle adjustment. Although the depth is available at the time of initialization in stereo SLAM, we add landmarks into the main filter in the same way by initialising landmarks through a separate information filter for improved accuracy.

5.5 Updating with Past Measurements

Our method can be thought of as a bundle adjustment within the Kalman filter, as we are updating the filter with all past observations of a newly added landmark. This is theoretically correct, as the Kalman filter can be updated either sequentially or as a batch (Evensen [2003]), as long as the same measurement is not used more than once to update the covariance of the filter (which makes the system over-confident).

If we retain k past measurements by setting the measurements $\tilde{\mathbf{z}} = \left[\mathbf{z}_t, \mathbf{z}_{(t-1)}, \ldots, \mathbf{z}_{(t-i)}\right]^{\mathrm{T}}$ and the Jacobian $\tilde{H} = \left[H_t, H_{(t-1)}, \ldots, H_{(t-i)}\right]^{\mathrm{T}}$ with measurement noise for all measurements $\tilde{\mathbf{v}} = \left[\mathbf{v}_t, \mathbf{v}_{(t-1)}, \ldots, \mathbf{v}_{(t-i)}\right]^{\mathrm{T}}$, The measurement equation in the previous chapter can be modified as follows:

$$\tilde{\mathbf{z}} - \tilde{H}\mathbf{x}'_{t_k} = \tilde{H}U_{s(t_k)}\mathbf{x}'_t + \tilde{\mathbf{v}} \tag{5.3}$$

In Equation 5.3, $\tilde{H}U_{s(t_k)}$ acts as a modified Jocobian $\tilde{H}_s$ for the reduced state. If the measurement vector size is n_m and the reduced state vector size is n_d, $\tilde{H}_s$ is a $n_m \times n_d$ matrix. Although this requires multiplying two matrices $\tilde{H}$ and $U_{s(t_k)}$), it can be done very efficiently as H is sparse. The reduced state size n_d is comparatively small, although the measurement size n_m can change. Multiplying both sides of Equation 5.3 by H_s^{T} gives the final reduced update equation which requires only inverting a $n_d \times n_d$ matrix.

5.6 Experiments

We proved the validity and accuracy of the proposed method by comparing it with global bundle adjustment and double window optimization. We used the publicly available New College data-set Smith et al. [2009] in our experimental set-up. Figure 5.1 shows the real-time results for a large-scale loop closure in that data-set. Unlike key-frame based SLAM, here we are updating all frames, which contributes to a much smoother trajectory.

Given that we collect enough information from the environment, the dimensionality of the problem in the reduced space can be kept very small irrespective of the original dimensionality. Usually the dimensionality can be maintained around

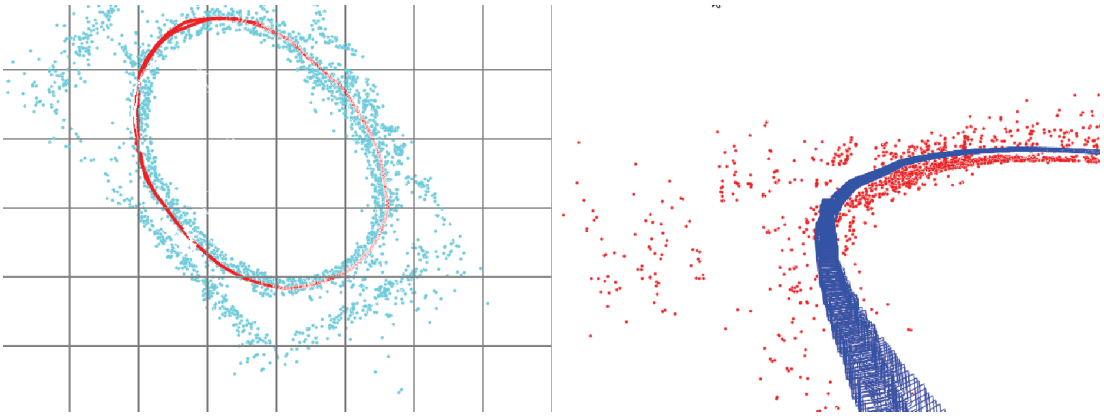


Figure 5.1: Large loop closure for stereo data.

20 for the New College data set as shown in Figure 5.2. With the number of observations, the dimensionality can be further reduced, as all observations contribute to reducing the same underlying structure of the problem with increased accuracy (it improves the signal-to-noise ratio).

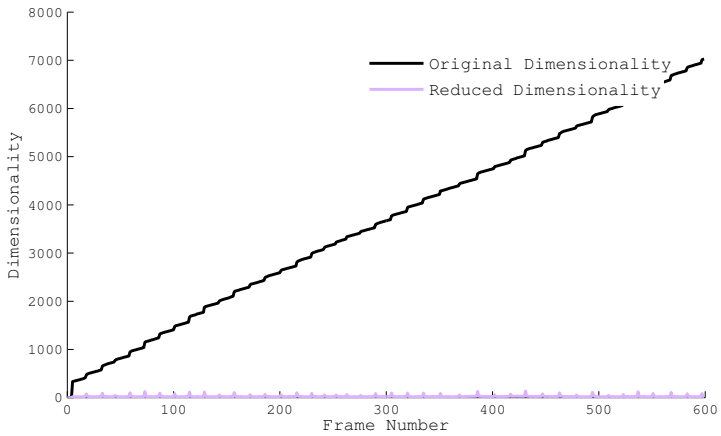


Figure 5.2: Dimensionality of the problem before and after the dimensionality reduction. The typical dimensionality after reduction is 20-30.

Furthermore, we compared the relative translation error for the pose estimates of the New College data-set. Figure 5.3 shows the relative RMS translation error. We compared the RMS translation error after normalising the translation to be unit vector against global bundle adjustment by treating global bundle adjust-

ment as the ground truth. It can be seen that our method yields better accuracy compared to double window optimization. This accuracy can be attributed to information fusion on every frame.

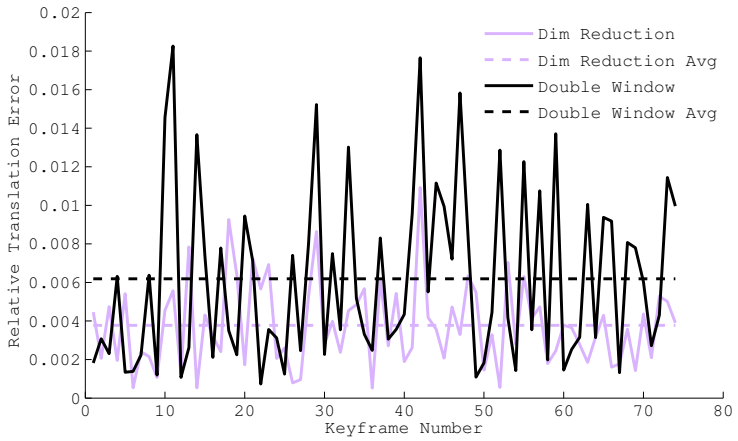


Figure 5.3: Translation error comparison between dimensionality reduced system and double window optimization

Next we compared the relative translation error of the monocular version of our systems with the stereo version. As shown in Figure 5.4, as expected, the stereo system yields far better results compared to the monocular version as the scale is readily available in a stereo system as the third measurement parameter. The scale drift is unavoidable in large scale monocular systems, necessitating the scale to be propagated through the loop after a loop-closure, as suggested in earlier work (Strasdat et al. [2010b]).

The most important factor to note is how dimensionality changes with the number of observations per frame. It can be seen that the number of significant dimensions changes with the number of observations per frame (Figure 5.5). The higher the number of visible measurements per frame, the higher will be the information available to recover the underlying structure with less uncertainty. This aligns with our intuition of reducing dimensionality along less uncertain axes of the system.

Figure 5.6 shows the experimental results obtained by rotating a camera on a

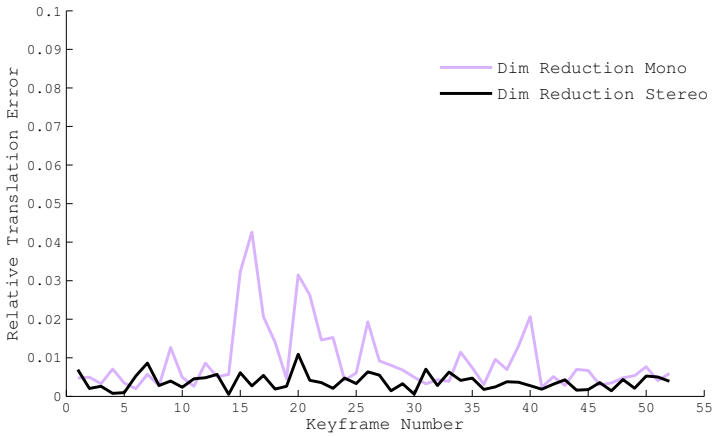


Figure 5.4: Stereo and monocular relative translation error.

turn table for an indoor sequence. The system can accurately close the loop for the given sequence.

5.7 Limitations and Further work

The proposed system produces estimates of all underlying state variables at each frame, rather than merely producing estimates of the observed landmarks. Here, the iterative dimension reduction step (performed each frame) remains the most expensive step of the system.

This means that our system complexity also grows with time and this limits speed. This limits our current implementation to medium scale SLAM - although it gives better estimation accuracy.

In our approach, although the update step requires only updating the reduced set of parameters, the expansion which maps the changes in the reduced space to the original space has to be done as a whole bundle. Here the older significant vector space $U_{s(t_k)}$ has to be multiplied by the newly-reduced vector space $U'_{s(t_n)}$.

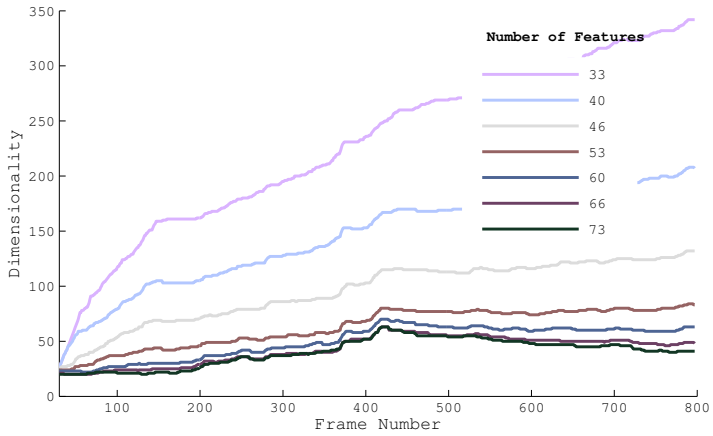


Figure 5.5: Dimensionality of the reduced system with the number of measurements

If there are n_L variables in the system state $\mathbf{x}_{t_n}$, the older reduced basis $U_{s(t_k)}$ becomes a n_L by n_D matrix if the reduce space dimensionality is n_D. Let us assume that the new reduction $U_{s(t_n)}$ is a n_D by $n_{D'}$ matrix. Then the complexity of this multiplication becomes $n_L \times n_D \times n_{D'}$.The number of variables grows rapidly, making this multiplication computationally expensive limiting its applicability to small to medium scale problems.

One of the solutions that we plan to investigate in future work is to update only the most recent set of variables by identifying a special structure of the significant vector space $U_{s(t_k)}$. When updating, this vector space can be modified using parts of $U'_{s(t_n)}$ related only to the most recent set of variables. However, when it is necessary to update the whole system, the multiplication has to be done over all variables, meaning that the complexity of the full problem remains the same. In addition, the reduction threshold has to be tuned by trial and error as it depends on factors like the rate of information acquisition and the filter parameters. If enough information is not received through measurements, it becomes harder to reduce the dimensionality.

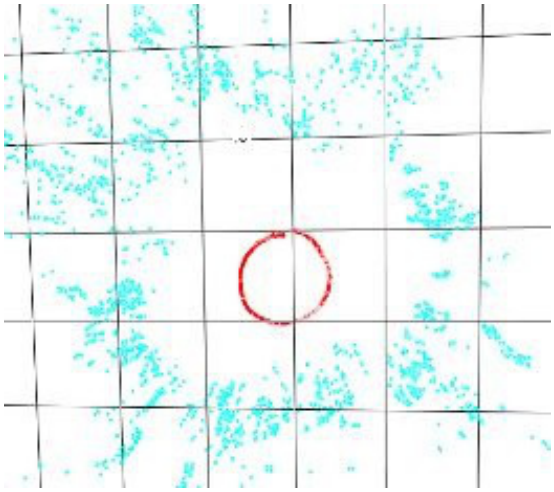


Figure 5.6: Trajectory of the camera for an indoor monocular sequence

5.8 Discussion

In this chapter we have introduced a dimensionality reduction technique for SLAM in a relative formulation. By identifying the most significant dimensions of the filter and updating only those dimensions we reduced the computational complexity. We applied the method to build a relative SLAM system which can solve small to medium scale problems accurately and efficiently.

This suggests that the underlying structure of the problem lies on a smaller dimensionality, although there are a large number of dimensions in the original problem.

We believe that our approach will shed some light upon the parameter selection of dense SLAM systems, to make it more efficient by identifying the underlying structure to be solved, without doing an exhaustive update.

Chapter 6

Monocular Image Space Tracking on a Computationally Limited MAV

In this chapter we present an efficient method for parallel tracking and mapping, enabling asynchronous communication between two threads. Our approach is derived from the recent development of parallel tracking and mapping algorithms, but unlike previous approaches, we show how the tracking and mapping processes can operate using different representations. The primary contribution is to show how the cost of tracking the vehicle pose on-board can be substantially reduced by estimating the camera motion directly on the image coordinate system rather than in the world coordinate frame. We demonstrate our method on an Ascending Technologies Pelican quad rotor, and show that the method is capable of tracking the vehicle pose with reduced on-board computation but without compromised navigation accuracy.

This work has been completed as a collaborative project with the **Center for Robotics and Intelligent Machines (RIM) at Georgia Institute of Technology** and the **Computer Science and Artificial Intelligence Laboratory (CSAIL) at Massachusetts Institute of Technology**.

In this work, our contribution is the front end Jacobian partitioning method. We identified that the measurement Jacobian can be partitioned into two parts. If we work in the pixel coordinate system, the Jacobian can be partially precalculated and saved as a database for fast on-line retrieval as shown below.

6.1 Introduction

Available SLAM algorithms are not suitable for deployment on limited platforms due to the computational cost. Here we are interested in monocular vision-based, inexpensive, and potentially disposable micro air vehicles (MAVs) that can be deployed in a large volume.

Although existing parallel tracking and mapping algorithms are capable of parallel processing, there is a tight coupling between these two operations. For instance, the PTAM (Klein and Murray [2007]) front-end requires estimates from its back-end in a synchronous manner, restricting the deployability of the front-end on a separate platform.

Solving the complete map is computationally demanding, especially as the number of landmarks grows large. PTAM addresses the computational cost by decomposing the computation into two parallel processes, such that a full-rate camera tracking process that uses the best available information and a map optimization process using a subset of representative camera images, i.e. key-frames. This decoupling requires sharing a single map in memory where the camera tracking process assumes a fixed map while in parallel, the map optimization process

continuously (if slowly) improves the map. Although proven very successful, this technique still carries the burden to optimise a growing global map. This difficulty is amplified in the kinds of low-power processors found in low-cost air vehicles that lack support for true parallelism.

An easy solution to reducing the cost of on-board computation is to move the entire pose computation to an off-board processor. Compressed images or feature-descriptor sets could be sent to a ground station and a pose estimate could be received in return. However, the wireless communication channel to an untethered MAV will typically suffer from packet drops, limited bandwidth and large transmission delays, that make time-critical dependence on the communication channel unreliable. A complete off-board scheme that includes such an unreliable communication channel in its control loop is not suitable for a MAV.

To overcome the communication limits while reducing the cost of on-board computation, only the costly map optimization can be moved to an off-board processor. A trivial solution is to send the entire map back and forth between the MAV and ground station, keeping different copies of the same information. However, once the on-board process is restricted to tracking, this process can be reformulated in the image space, leading to much reduced computation. The mapping process can continue to run off-board, and provide asynchronous copies to the MAV of the updated map as the communication channel permits, where the updated map is projected into image space and bounded by visibility in the current frame, specifically for the purposes of fast, on-board pose estimation.

The main contribution of this work is the reformulation of the tracking and mapping problem when the two processes are physically decoupled and no longer share the same copy of the map. We introduce monocular image space tracking (MIST) and show that the tracking process can be entirely in the image coordinate frame for fast computation. We discuss landmark representation in the image space, data association using such landmarks, fast pose optimization using pre-computed Jacobians, updating the landmarks frame-to-frame, and forward projection to compensate for the delay in asynchronous updates from the ground

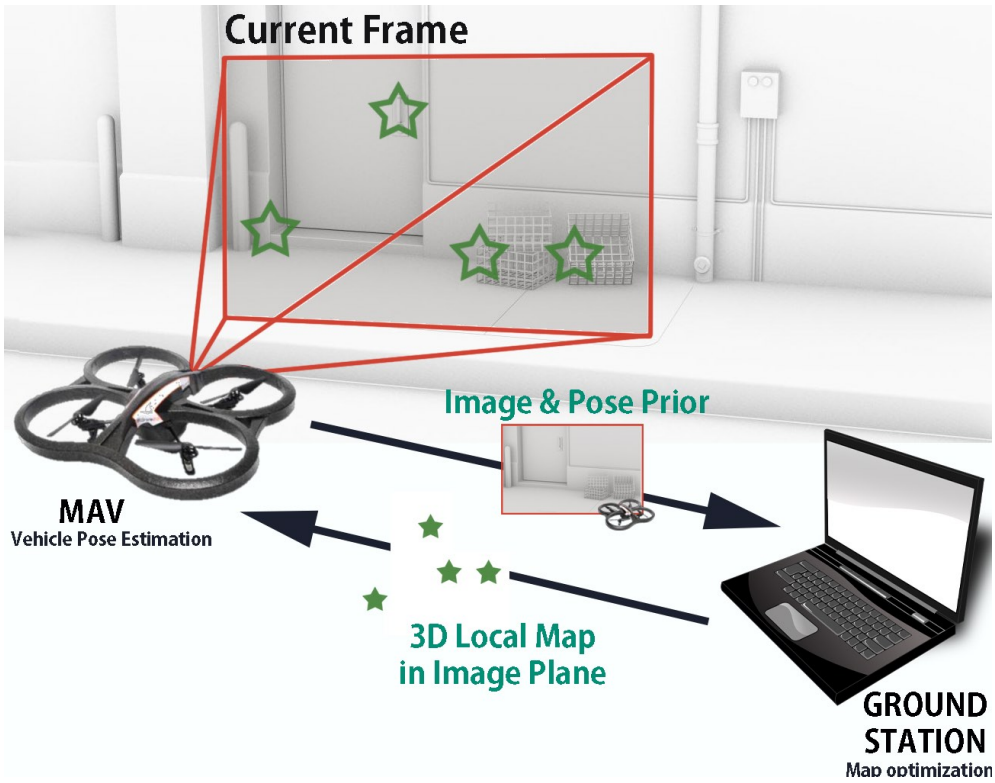


Figure 6.1: The MAV occasionally sends the camera image and its own pose estimate to the ground station for map building. The ground station sends back a local map in the image space of the MAV for fast pose tracking.

station.

Our novel approach of asymmetrically distributing SLAM onto separate devices results in a fully scalable robust pose estimation on a computationally and bandwidth limited MAV, while the globally consistent map of arbitrary size can still be inferred on the ground. We demonstrate the improvements achieved with MIST using a monocular camera mounted on an Ascending Technologies Pelican quad-rotor.

6.2 Background

In the field of autonomous navigation and exploration, the use of planar laser scanners have shown success in achieving full autonomy for micro air vehicles (Bachrach et al. [2011] Shen et al. [2011]). However, these methods do not scale well into inexpensive platforms due to the weight, power requirement, and the cost of laser scanners. Recent work (Shen et al. [2013], Fraundorfer et al. [2012]) has demonstrated similar exploration capabilities using stereo vision as primary sensing means. Although abandoning the laser scanner reduces the weight and power requirements, stereo vision techniques are still computationally expensive. We would like to push towards the limits of minimal sensor suite and minimal processing efforts by utilising a monocular camera.

Past work on the use of a monocular camera on a MAV for on-board SLAM (Weiss et al. [2011]) has used PTAM as a black-box pose estimation unit. However, originally developed for augmented reality applications in small workspaces where the explored map is assumed to be small, PTAM's computational requirements would still be a burden on computationally limited MAV platforms exploring a larger area.

Recent work by Forster et al. [2014] reduced the processing requirements

greatly, and demonstrated robust high frame-rate tracking using a small processing unit. Although their SVO method has low frame-to-frame computational requirements, it is not fully scalable due to the burden of storing a growing map in the MAV's memory. In addition, SVO requires a two-core processor for optimal speed for its parallel tracking and mapping design; our system requires only a single core on the MAV with all of the heavy computational load on the ground station. Lastly, SVO is currently engineered to use a downward camera. Therefore, an algorithm that is as fast on a single core processor with a small memory capacity, that could work with both downward and forward cameras would be much more desirable.

As opposed to on-board methods, some previous work (Ok et al. [2012], Engel et al. [2014b], Ta et al. [2014]) has streamed images from the MAV to a more capable ground station to off-board the computations. However, this strategy requires aggressive image compression and reduced frame rates, leading to overall poor image quality. Computing the pose of the MAV on a ground station, and streaming it back to the MAV also introduces a large transmission delay in the pose updates needed by the on-board controllers. There are techniques for mitigating the controller errors that can result from a delay in the state estimate (Engel et al. [2014b]), but these solutions are ultimately not as robust as a high-rate on-board state estimation process.

Other previous work (Ni et al. [2007], Frese [2006], Williams et al. [2002]) partitions the SLAM problems to meet different objectives, but our work is novel in dividing the problem onto two separate devices to meet the requirements of a computationally limited system.

Following the same notation used in Chapter 4, let x_{t_k} represent the state of the system which includes the camera parameters $x_{c_{t_k}}$ and landmarks $x_{y_{t_k}}$. In a fully decoupled system, the limited platform should receive landmark estimates visible on a given camera frame which can be regarded as a local map, which may not necessarily be the current frame due to the asynchronous nature of communication. The MAV, should compute the current camera pose $x_{c_{t_k}}$ given

the current image, and the transmitted local map of landmarks in the pixel space.

Conversely, on the ground station, we take the smoothing and mapping (SAM) approach (Dellaert and Kaess [2006]) to solving the SLAM problem. On the ground station we optimise over a selected subset of key-frames that are far apart from each other. We do this by occasionally receiving camera images I_t from the MAV, and optimising for the maximum a posteriori (MAP) estimates.

6.3 Computation on the MAV

This section explains the main contribution of this chapter. Here we propose a novel Jacobian partitioning method which enables the pre-calculation of a normalised version of the Jacobian for later direct retrieval. This relieves the computationally limited MAV from computing the Jacobian each time.

We adopt a feature-based approach, where we extract a set of features $x_y t_k$ from a given image. Those features are matched against the next image for feature correspondences.

When the landmarks are represented relative to some coordinate system, to do the feature-landmark association those landmarks have to be projected onto the current frame through the kinematic chain. As the system has two threads which work independently, we delegate most of the computations to the server. We are able to save computations by having the ground station do the initial projection for the MAV and creating a local map in the image coordinate system. To do this, the ground station optimises the global map using the latest information received from the MAV, from which it can maintain a refined $3-D$ estimation of landmarks. Then the ground station can predict the next frame and project $3-D$ landmarks in the global map onto this frame to create a local map of landmarks $\bar{L}_{t_k}$. The ground station also discards landmarks that are out of view, so that the

matching on the MAV uses only a small local map usable in the near-future. Due to the asynchronous communication that we assume between the two platforms, the local map $\bar{L}_{t_k}$ sent by the ground station will only be available to the MAV at a some later time t. Therefore, after receiving a map from the ground station, the MAV has to project the local map $\bar{L}_{t_k}$ onto the current frame at time t to obtain an updated local map $\bar{L}_t$.

In our implementation, we extract FAST (Rosten and Drummond [2006]) corners at 4 pyramid levels and store them in a grid at each pyramid level. We use the grids to reduce the number of potential matches before comparing the actual descriptors or enforcing the maximum feature distance in the image space.

6.3.1 Forward Projection

As we allow the MAV to be autonomous, by the time it receives the local map $\bar{L}_{t_k}$, it may have proceeded a few frames further, as mentioned earlier. In order to bring this map to the current camera frame of the MAV, the received local map $\bar{L}_{t_k}$ has to be projected through the kinematic chain from frame t_k to t. The MAV can linearly project this received map into its current view by composing the kinematic chain. We keep computing the kinematic chain using the feature correspondences that MAV received from the ground station as $\bar{L}_{t_k}$. This forward projection frees the MAV from time consuming computations, reducing the risk of crashing with temporary losses in the communication channel.

Let the current frame of the MAV be at the time step t and t_k be the time step where the projections $\bar{L}_{t_k}$ from the ground station are available. The forward projection will bring this map onto the current frame as $\bar{L}_t$. This can be easily done using between frame feature correspondences.

If the number of visible landmarks of $\bar{L}_{t_k}$ goes below a threshold, we also initialise a new set of landmarks on-board, where the depth of such landmarks

can be roughly estimated.

Given the measurement Jacobian, the incremental transformation between frame i and j such that $j > i$ can be easily calculated as follows:

$$h(x_{t_j}) \approx h(x_{t_i}) + H_i \xi_{ij} \tag{6.1}$$

where, h is the measurement function. Since landmarks are directly represented in pixel coordinates, it is not necessary to do a $3-D$ to $2-D$ projection in the measurement function. The Jacobian is directly looked up from a database, as discussed later. If more computational power is available the estimation accuracy can be improved using the 4^{th} order Runge-Kutta method to update the inverse distance Q_j from frame to frame. However, this update is not crucial since the inverse distance does not change significantly between a few frames. We choose the inverse distance representation over the inverse depth counterpart for this reason.

6.3.2 Motion Calculation and Outlier Rejection

Given the data correspondences $C_{(t,t-1)}$, we can calculate the pose x_{c_t} by maximising the posteriori probability:

$$P(x_{c_t}|C_{(t,t-1)}, \bar{L}_{t-1}) \propto P(x_{c_t}|\bar{L}_{t-1}) \prod P(z_t^j|x_{c_t}, x_{y_t}^j) \tag{6.2}$$

where z_t^j is the corresponding image feature of the landmark $x_{y_t}^j$. Assuming a Gaussian priori on the pose x_{c_t} and Gaussian measurement noise, where R is the covariance matrix, this is equivalent to minimising the negative log-likelihood:

$$\arg\min_{x_{c_t}} \parallel x_{c_t} - x_{c_{(t-1)}} \parallel_{\Sigma}^2 + \sum \parallel z_t^j - h(x_{c_t}, x_{y_t}) \parallel_R^2 \tag{6.3}$$

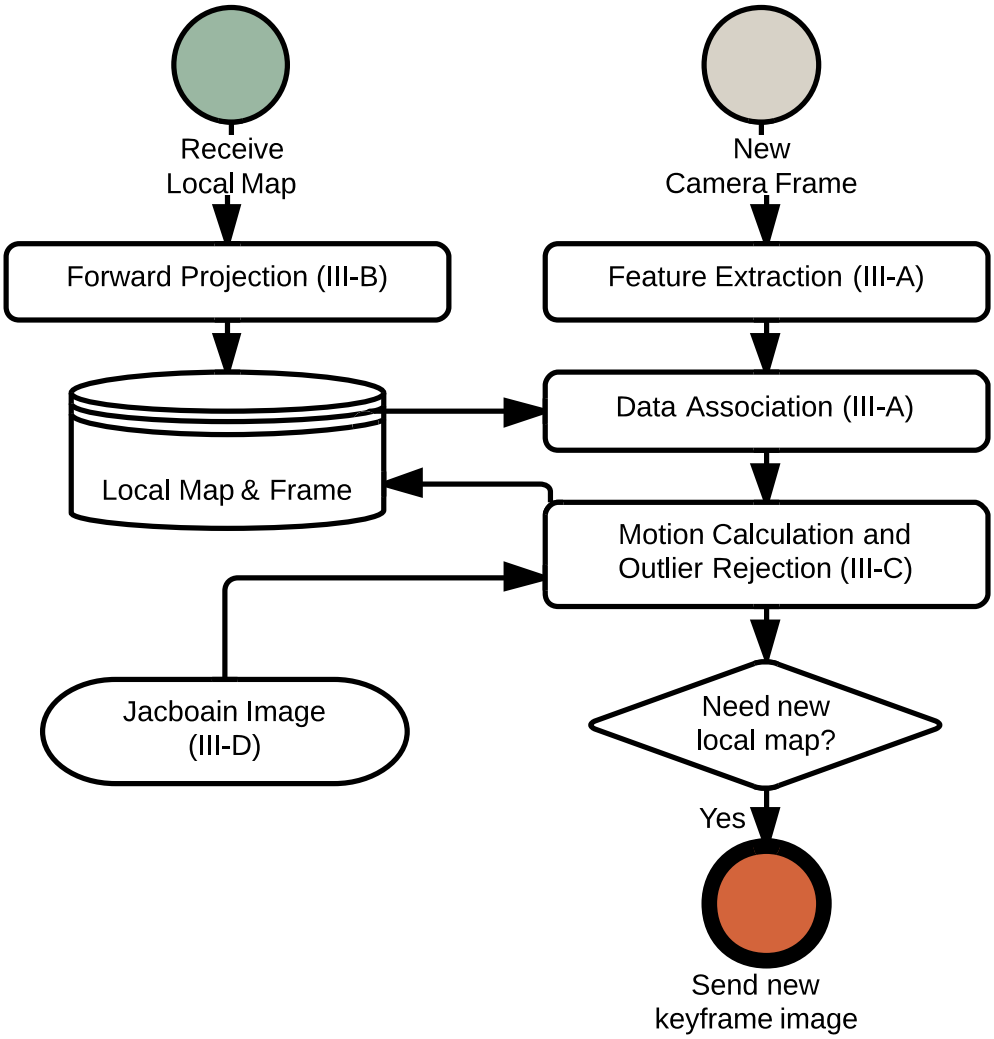


Figure 6.2: Flow diagram of operations done on the MAV.

where $h(x_{c_t}, x_{y_t})$ is the measurement function, and z^j is the measurement of a landmark, i.e. the associated feature location.

The current pose x_{c_t} can be parametrised as an incremental change from the previous pose $x_{c_{t-1}}$ such that $x_{c_t} = exp(\xi) \oplus x_{c_{t-1}}$, where $\oplus$ denotes pose composition in $SE(3)$. Hence, with linear approximation, the observation model becomes

$$h(x_{c_t}, x_y^j) = h(exp(\xi) \oplus x_{c_{t-1}}, x_y^j) \approx h(x_{c_{t-1}}, x_y^j) + H_j \xi \tag{6.4}$$

where H_j is the 2×6 Jacobian matrix defined by:

$$H_j = \frac{\partial h(x_{c_t}, x_y^j)}{\partial x_{c_t}} = \frac{\partial h(exp(\xi) \oplus x_{c_{(t-1)}}, x_y^j)}{\partial \xi}|_{\xi=0} \tag{6.5}$$

Let x_y^j be a local landmark. In the MIST framework, we represent local landmarks on the image space as $x_y^j = (u^j, v^j, Q^j)$, where $p^j = u^j, v^j$ are the pixel coordinates. Therefore, our tracking problem can be posed as a *linear* optimization:

$$x_{c_t}^* = \arg\min_{\xi} \parallel x_{c_t} - \tilde{x}_{c_t} \parallel_{\Sigma}^2 + \sum_{j=1}^{K} \parallel (z^j - p^j - H_j \xi) \parallel_R^2 . \tag{6.6}$$

where $\tilde{x}_t$ is the latest state estimate.

Since there is no perfect data association in practice, we iterate the computation a few times by re-weighting R based on the residuals. This iteratively re-weighted least squares (IRLS) framework (Paul and Roy [1977]) serves two purposes: 1) it reduces the effect of outliers and 2) the final weights can be used to evaluate the quality of pose tracking. This quality assessment along with the percentage of successful data associations are used to judge whether a new frame should be sent to the ground station for a subsequent local map update.

6.3.3 Jacobian Image

What enables pre-calculating the Jacobian is the special structure identifiable in the Jacobian matrix H_j used in Equation 6.6. The Jacobian on the pixel coordinate system can be formulated as:

$$H_j = \left[\frac{\partial I}{\partial p}\right] \begin{bmatrix} Q & 0 & -QU & -UV & 1+U^2 & -V \\ 0 & Q & -QV & -1-V^2 & UV & U \end{bmatrix} \tag{6.7}$$

where $p = (U, V)$ are normalised camera coordinates $(x/z, y/z)$. The first three columns of the H_j correspond to translation parameters and the second three columns correspond to rotation parameters of ξ, where $I = (u, v)$ are the pixel coordinates. Thus, the first term, $\frac{\partial I}{\partial p} = \begin{bmatrix} \frac{\partial u}{\partial U} & \frac{\partial u}{\partial V} \\ \frac{\partial v}{\partial U} & \frac{\partial v}{\partial V} \end{bmatrix}$, depends on the calibration model of the camera.

In Equation 6.7, if we divide the first 3 columns of H_j by Q, the result only depends on the pixel location (where U, V are functions of u, v). Therefore, we can pre-calculate this matrix at every pixel and store the matrix as an image of 2×6 matrices that can be used at run-time to reconstitute the Jacobian quickly from a pixel coordinate and Q_t^j for any landmark by retrieving the 2×6 matrix at p_t^j and multiplying the first 3 columns by Q_t^j.

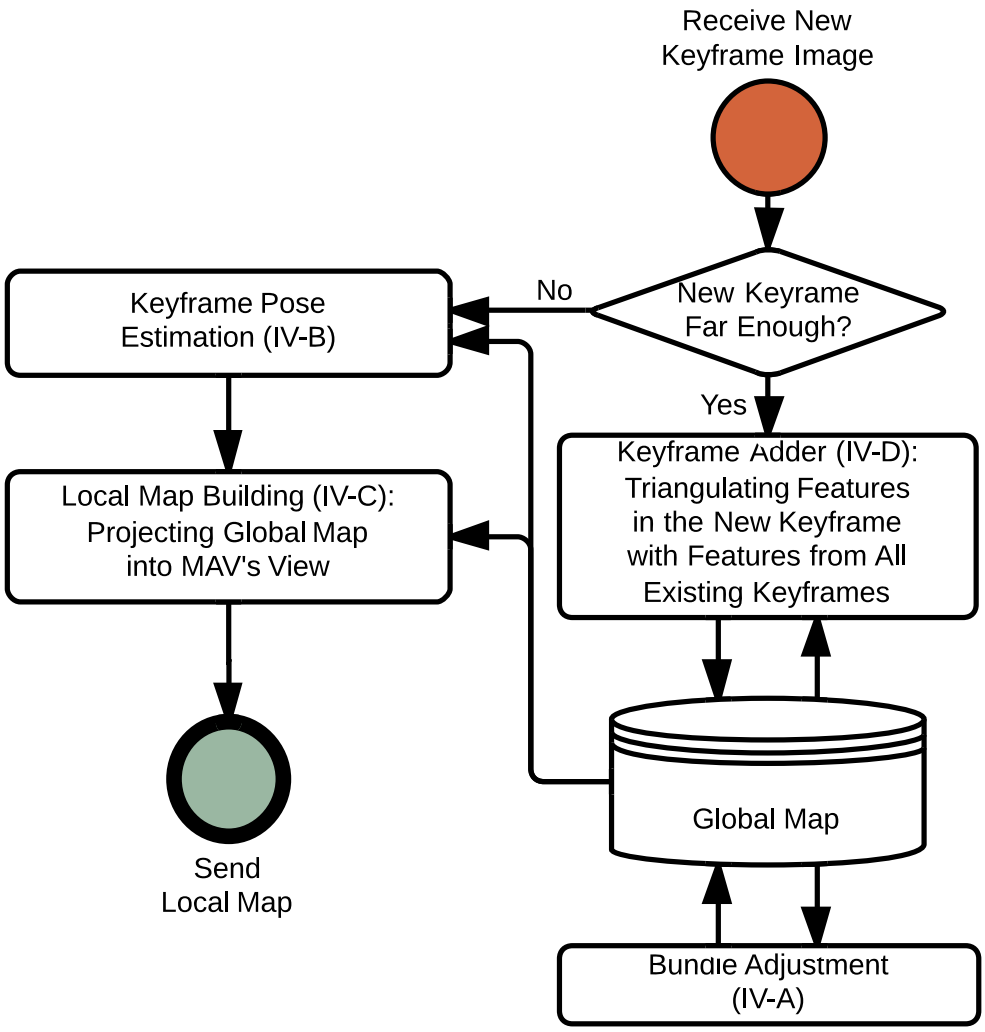


Figure 6.3: Framework showing the ground's operations in relation to the MAV's.

6.4 Computations on Ground Station

6.4.1 Bundle Adjustment

To distinguish between the computations of the ground station from those of the MAV, let us use X to represent the state. Here we drop the time dependence t for clarity, assuming state to be current. The ground station computes a global map of landmarks and key-frame camera poses X given visual measurements by recovering the (MAP) estimate:

$$\begin{aligned} X^* &= \underset{X}{\operatorname{argmax}}\, P(X|Z) \\ &= \underset{X}{\operatorname{argmax}} \prod_i P(x_c^i) \prod_{i,j} P(z_{ij}|\, x_c^i, x_y^j) \end{aligned}$$

This map-building problem can be posed as inference on a factor graph (Dellaert and Kaess [2006]). The variable nodes are camera poses x_c^i and the landmarks x_y^j while the factor nodes are the prior densities $P(x_c^i)$ on the variable nodes, and the measurement likelihoods $P(z^{ij}|\, x_c^i, x_y^j)$ constraining a pose x_c^i and a landmark x_y^j, given the corresponding visual measurement z^{ij}. This measurement likelihood is equivalent to the observation model used on the MAV, described in Equation 6.4. By eliminating the factor graph, we can solve for all the camera poses and the landmarks. We omit the details of this process, since we use standard inference techniques for the camera poses and landmarks in the world co-ordinate frame, rather than inference in the image frame as we do on-board the vehicle.

6.4.2 MAV Pose Estimation

Parallel to the process of bundle adjustment, the ground station periodically receives images from the MAV. We perform a similar pose estimation to that done on the MAV in the standard world co-ordinate frame to estimate the pose of the image. This pose optimization is the well-known camera re-sectioning problem, i.e. computing the optimal camera pose $x^*_{c_t}$ given measurements z^j_t of *known* landmarks:

$$\begin{aligned} x^*_{c_t} &= \underset{x_{c_t}}{\operatorname{argmax}} P(x_{c_t} | \{z^j_t, x^j_y\}_{j=1..m}) \\ &= \underset{x_{c_t}}{\operatorname{argmax}} P(x^i_c) \prod_j P(z^j_t | x_{c_t}, x^j_y) \end{aligned}$$

where we use the MAV's pose estimate as a prior $P(x^i_c)$.

One thing to note is that the MAV and the ground station can use different corner features and feature descriptors, since the MAV pose tracking is repeated on the ground station. Accurate but computationally expensive methods such as SIFT (Lowe [2004]) can be used on the ground station in place of the lightweight algorithms on the MAV.

6.4.3 Projection to View

The ground station can apply the MAV's camera model to obtain estimated *image coordinates* p^j on MAV's image space for each probably-visible landmark and communicate these rather than the metric locations. Furthermore, at any given time step, the MAV only needs to know about the landmarks that it is likely to observe, and the ground station therefore only needs to transmit a map of these landmarks.

Therefore, using the optimised MAV pose, we project the landmarks x_y onto the calibrated coordinates (U^j, V^j):

$$(U^j, V^j) = K[R^c_w | t^c_w] x^j_{yw}$$

and then to in-image *pixel coordinates* $(u^j, v^j) = f(U^j, V^j)$ using f the fish-eye lens model (Devernay and Faugeras [2001]). Out of these projections $p^j = (u^j, v^j)$ in pixel coordinates, those within the image boundaries are included as a local map. This map is then transmitted to the MAV, as illustrated in Figure 6.3.

6.4.4 Adding Key-Frames

The two initial frames are created in a separate initialization process using homography with a locally planar assumption, as in Klein and Murray [2007]. During the initialization stage, all of the frames are transmitted from the MAV to track a trail of features on the ground station. After initialization, the ground station waits for the MAV to send a new frame, while optimising the global map in parallel. Once a new frame is received, the ground station prepares a new local map in the image plane of the frame and sends it back to the MAV. Then the frame's distances to all other key-frames in the global map are calculated to judge whether it is far enough from other key-frames to qualify as a new key-frame. The ground station then searches through all of the known key-frames to make data associations with the newly received frame. This dense association stage is critical to building a globally salient map.

(a) Figure A

(b) Figure B

Figure 6.4: The MAV uses a monocular camera and tracks landmarks in the image space to estimate its pose

6.5 Experiments

We autonomously flew an Ascending Technologies Pelican quad-rotor in an unknown indoor environment, using a 30 fps PointGrey Chameleon camera, a Microstrain IMU, a Hokuyo laser scanner, and a Gigabyte dual-core i7 to evaluate MIST as a visual pose estimation module on a MAV. We then analysed the accuracy of our pose estimates by comparing them with the pose estimates generated using PTAM. We also compared the time to pose estimation for MIST, PTAM running on-board, and PTAM running off-board. The dataset for benchmarking is collected by hand-carrying the quadrotor around an indoor environment, and saving camera images and other sensor data using LCM (Huang et al. [2010]). The saved data is played back at the original intervals on the quad-rotor to simulate the MAV flying, while providing the exact same input to different algorithms used in comparison. For the ground-station, we use a quad-core i7 laptop.

6.5.1 Autonomous Flight using MIST

During the autonomous flight, we ran a laser-scan matching algorithm (Bachrach et al. [2011]) in parallel. The pose estimates and the occupancy grid from the laser scan-matcher were treated as ground truth; we obtained metric pseudo-scale input from the pose estimates, and used the occupancy grid to plan a collision-free trajectory, as shown in Figure 6.5. The use of the laser-scanner was for these purposes only, and our algorithm does not require the laser to estimate the pose.

We formed the flight trajectory by clicking waypoints and using a polynomial trajectory generator (Richter et al. [2013]) to smoothly connect them within the laser-built occupancy. We controlled the quad-rotor using a nonlinear $SE3$ controller (Lee et al. [2010]) and increased the frequency of the vehicle pose estimate by relying on an EKF to fuse in our MIST vision estimates and a 100 hz IMU.

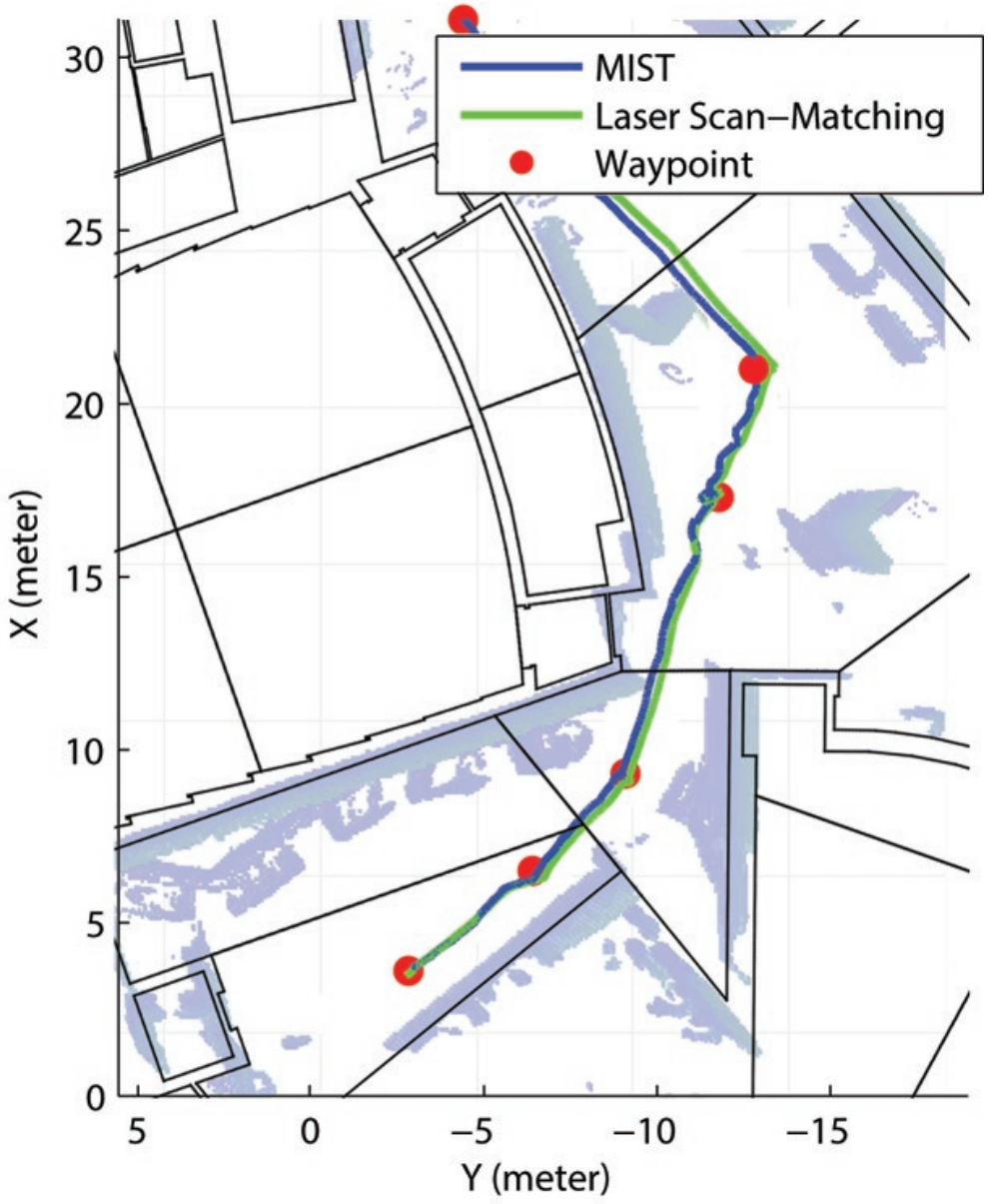


Figure 6.5: Waypoints are manually clicked to generate a trajectory, and MIST is used to estimate the pose and follow the trajectory. Laser scan-matching results are run in parallel and used as the ground-truth.

In order to align the laser-based poses and the occupancy grid with the vision-based poses, we transformed the vision-based pose updates in the camera frame to updates in the robot body frame, and continuously composed to an initial laser pose. To resolve the scale ambiguity, the vision-based poses and the laser-based poses were collected between two consecutive local map updates and the difference in translation is used as a pseudo-scale input to re-scale the map for the following frames. An alternative to the metric scale estimates from the scan-matching could be integrating the accelerometer (Weiss and Siegwart [2011]), or using a single-point range-finder (Engel et al. [2014b]).

As shown in Figures 6.5 and 6.6, while the MAV can reliably follow the trajectory, continuously composing the vision-pose updates on a single initial laser pose results in the vision-based trajectory eventually drifting away from the laser-based trajectory as errors accumulate. One source of systematic error between the vision-based estimates and the laser-based estimates is the approximate transformation between the camera and the laser frame, caused by an approximate hand-alignment of the camera, the IMU, and the laser on the quad-rotor. Another source of approximation error is the linear interpolation performed on the laser pose estimates when finding the pose chronologically closest to a camera time-stamp. The last source of error is the heavy dependence on the pseudo-scale inputs, which corrupt the translation if estimated poorly.

6.5.2 MIST Tracking Accuracy

To evaluate the accuracy of MIST in the presence of external systematic errors, we collected camera images and sensor data and played them back as identical inputs to PTAM and MIST. Over a 50 meter trajectory shown in Fig. 6.6, our pose estimates were approximately as good as the trajectory generated using PTAM.

We compared the error in rotation and translation of frame-to-frame updates

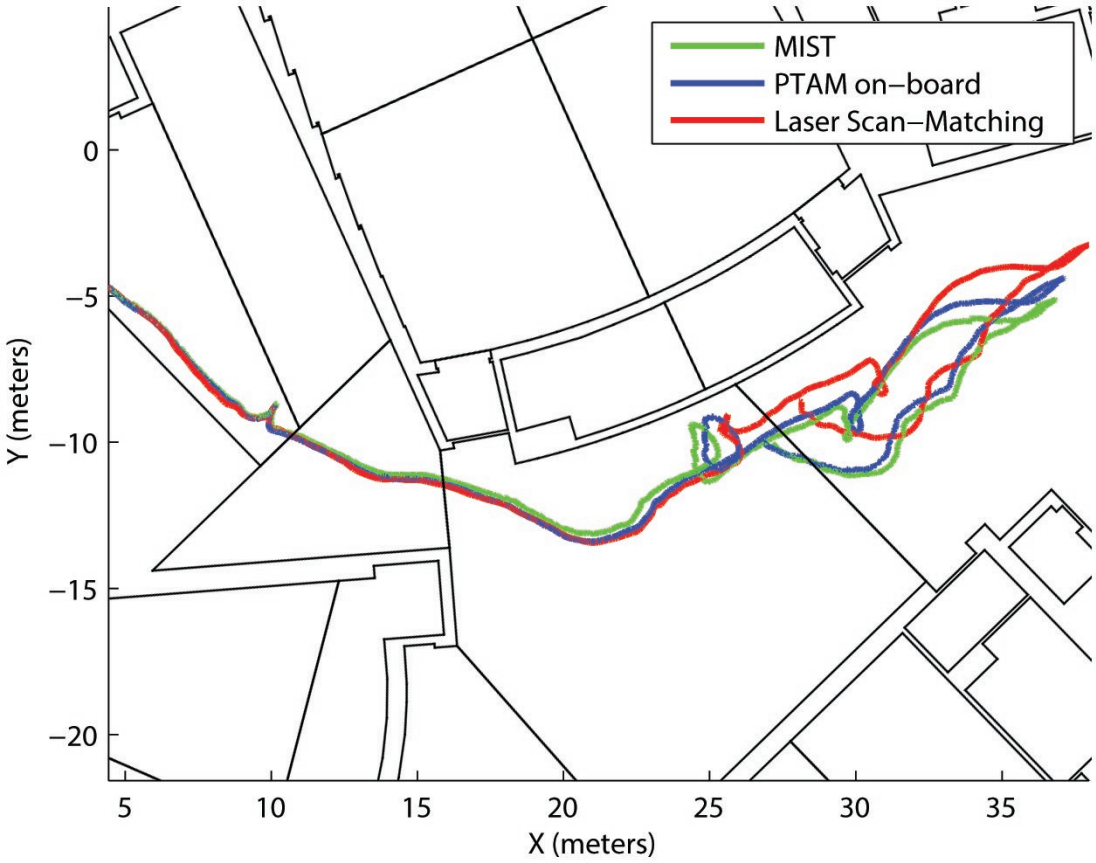


Figure 6.6: The laser-based trajectory estimate in red, our pose estimates in green and the PTAM estimates in blue. During the total travelled distance of 50 meters, the vision-based trajectory drifts aways from the laser estimates. However, our method is approximately at least as accurate as PTAM.

in the robot body frame for PTAM and MIST, with the laser scan-matching as the ground-truth. It can be seen in Figure 6.7 that the performance of our system is nearly identical to that of PTAM at approximately less than 0.4 degrees of error in each of roll, pitch, and yaw while the error in translation is under 3 cm.

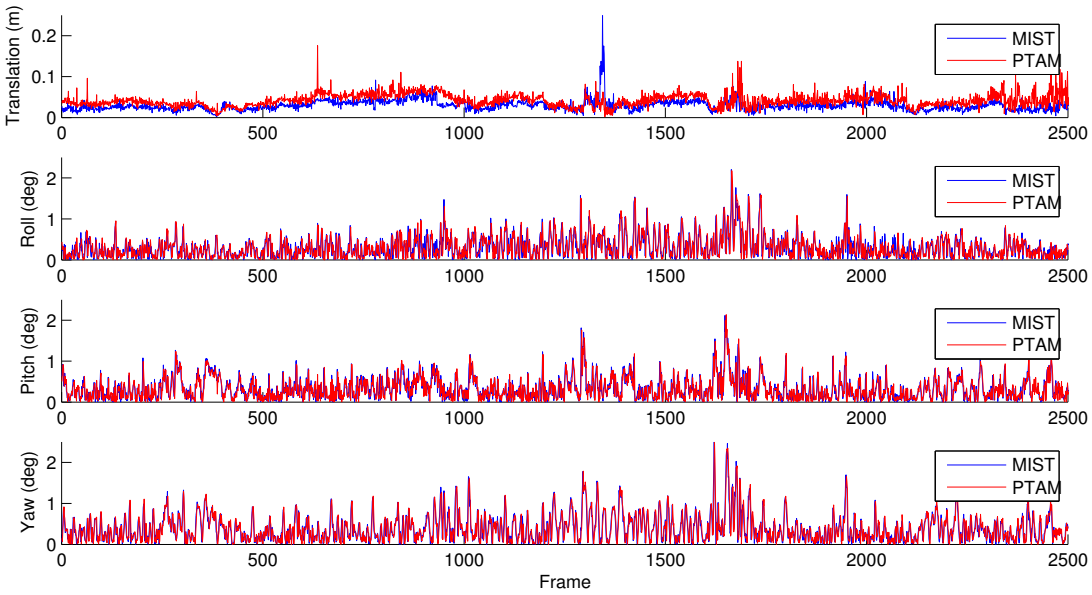


Figure 6.7: Errors in translation and rotation for PTAM and MIST, compared to the laser scan-matching counterpart. It can be seen that the accuracy of MIST is on par with that of PTAM.

6.5.3 Timing Comparisons

We compared the time to pose estimates between our system, PTAM running on-board, PTAM with raw streamed images, and PTAM with JPEG-compressed streamed images. As with the trajectory analysis, pre-recorded LCM messages are played back on the MAV to simulate the vehicle flying while providing identical sensor data to different methods. In the case of streaming methods, the time to pose estimate includes the round-trip transmission time over wi-fi as well as the pose computation time on the ground station. For MIST and PTAM running on-board, this time is only the computation time taken since the camera image was available.

As shown in Figure 6.8, sending uncompressed images takes over 200 ms, causing the pose estimates to arrive about 8 frames later. This large delay in transmission makes tracking infeasible, as any rapid motion causes PTAM to lose

tracking. On the other hand, streaming JPEG images is relative fast, with the estimates lagging two frames behind. Although this delay could be moderate for slow moving vehicles, the transmission delay is not always constant; there are regions where the delay is more than double, and if a few packets are dropped at any point during the entire motion, it would cause PTAM to lose track and cause the vehicle to crash.

In this figure, it can be observed that the time taken for PTAM and MIST do not differ much. The reason is that on the dual-core i7, the computation done by PTAM's mapping process is in fact truly parallel to the tracking thread, and does not add to the computation time to pose estimate. While this shows the strength of the parallel design, we emulated a more computationally limited platform, typically found on low-cost MAVs, by enabling only a single core. We also quadrupled the play-back speed of the LCM messages so that the camera images come at 120 hz, while the IMU comes at 400 hz.

As shown in Figure 6.9 on an emulated single-core machine, MIST still retains a constant time to pose estimates. However, for PTAM running on-board, as the processor jumps from the tracking process to the mapping process, the time to pose estimate grows with the growing map, due to increasing difficulty in bundle adjustment. We also disabled the entire mapping process of PTAM, and run only its tracker to compare against MIST. The average computation time for MIST is 3.97 ms, capable of estimating the pose at over 200 hz, while the average for PTAM tracker is 7.78 ms. It can be seen that the computation time for the PTAM tracker also grows slowly since it has to project the growing map into its measurement space.

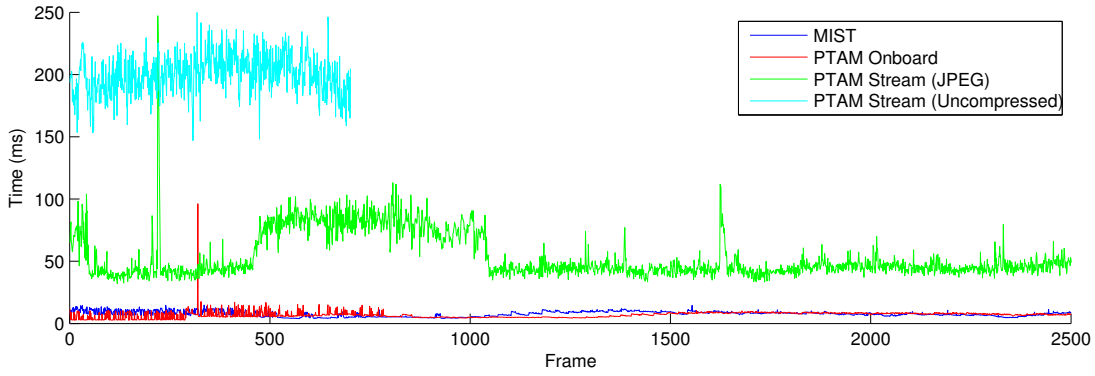


Figure 6.8: Comparing the time to pose estimate for MIST, PTAM running on-board, PTAM with streamed images. It can be observed that it is infeasible to track using raw streaming method, while the JPEG-compressed images still take more than a full frame to arrive at the MAV.

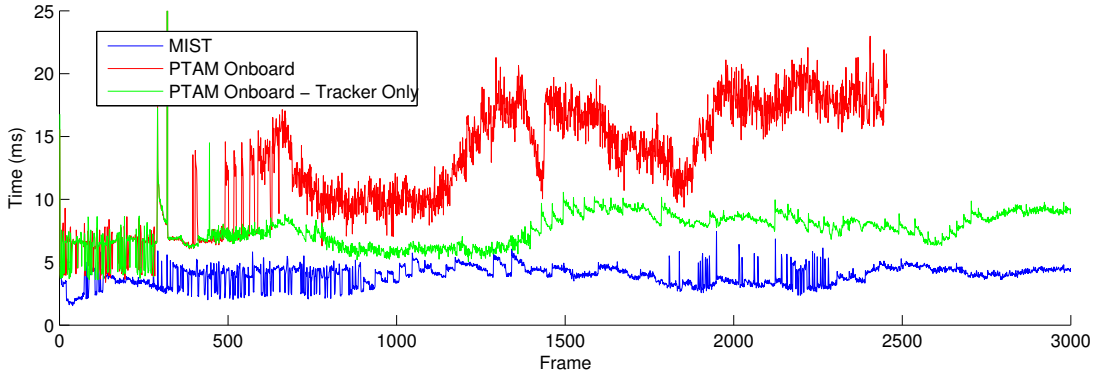


Figure 6.9: A single-core computationally limited system is emulated to better highlight the difference between PTAM running on-board and MIST. It can be observed that the computation for our tracker remains constant, while PTAM gradually increases time due to the mapping process. Comparing against only the tracker process in PTAM, MIST is nearly twice faster.

6.6 Discussion

We have divided the full SLAM problem into fast monocular image space tracking (MIST) on the MAV and key-frame-based smoothing and mapping on the ground station. Using our fully decoupled tracker and mapper design and fast image space tracking, we are able to compute the pose estimates on the MAV in constant time while building the growing global map on the ground station. The quality of this global map is as accurate as PTAM, as shown by the the pose estimates of MIST and PTAM compared against laser scan-matcher ground-truth.

In future work, we can explore utilising MIST in a multi-robot scenario where a team of disposable MAVs are needed to navigate a hazardous area. Having a single ground-station with multiple low-cost MAVs sharing a single map may be a trivial solution to creating a centralized distributed system.

Chapter 7

Conclusion

This thesis mainly focused on reducing the dimensionality of the extended Kalman filter for SLAM. As information is learnt about the environment, the covariance matrix in the Kalman filter typically becomes dense, with strong correlations between the locations of individual landmarks. As a result the covariance matrix becomes spectrally sparse (with large numbers of small singular values). We proposed an algorithm that takes advantage of this by removing dimensions from the filter about which we have learned "enough, meaning that their singular values have become sufficiently small that they are known to the system to a high precision. This is achieved by maintaining an orthogonal mapping between the full-dimensional underlying state and the reduced set of dimensions. Results show that this method yields estimates for the camera pose and map, that are nearly identical to those of the standard Kalman filter, while reducing the computational cost. In the initial implementation we marginalized out previous camera poses from the filter, only maintaining the current pose estimate following the conventional EKF-SLAM.

Then we extended the proposed reduced dimensionality Kalman filter in to a

relative formulation, where each landmark is represented relative to its initialized camera frame, giving a more linear solution to the problem inspired by the relative bundle adjustment for SLAM. Because of this high linearity, we showed that the system can be extended to larger environments. Also we proposed a method to absorb more information into the main filter by introducing measurements as a batch when adding landmarks, those maintained in independent filters until the depth is estimated to a sufficient accuracy. The proposed dimensionality reduced system can work as an efficient back-end for small to medium scale SLAM. When the scale of the problem grows, the frame rate starts dropping as we are propagating the updates in the reduced space to the original space updating all parameters of the system.

Also in-order to make the front-end more robust towards noise, in this thesis we proposed a novel frequency domain sub-pixel refinement technique for corner match refinement in SLAM, extending the spatial domain Gauss-Newton affine parameter estimation into frequency domain. Our approach draws a relationship between the Gabor filter frequency response and the frequency response of a Gaussian multiplied patch. Using this relationship, we showed how to select the correct frequency range for parameter estimation, eliminating the DC response and aliasing.

Finally we extended parallel tracking and mapping (PTAM) by Klein and Murray [2007] to combine a SLAM front-end which can work asynchronously extracting and tracking features, with an optimization back-end. Being different from the existing PTAM approach, our method enables asynchronous communication between the two threads, so the front-end can work independently as a standalone entity for some period of time until information from the back end becomes available. We showed how the tracking and mapping processes can operate using different representations, reducing the computational cost of using the same model for both, and allowing the two models to be updated asynchronously. Because of the asynchronous communication, our front end can be deployed on a separate platform like a quad-rotter which can communicates with a back-end server for path correction information.

Though we have presented our reduced dimensionality EKF as a solution to optimize all SLAM parameters, we believe that it would be much useful if used as a local solution. For instance when applied to a graph based SLAM system like coalesced observations SLAM presented by Eade and Drummond [2007], we can reduce the dimensionality of each node, intern increasing the number of landmarks each node can handle or it can be used in semi-dense systems to increase the number of pixels used. Especially in a double window approach as described by Strasdat et al. [2011], the method can be used to handle the inner window.

Feature based SLAM has become highly accurate due to recent developments in algorithms and processing technology. Most recent dense methods show the possibility of improving this accuracy further by dense reconstruction of the environment, which effectively improves the signal to noise ratio of the system. Still the dense systems are not capable of handling all parameters jointly. We believe that our method will shed some light upon enabling joint optimization of landmarks by identifying the significant set of internal parameters of the system.

Appendix A

Mathematical Framework

This chapter presents the mathematical notation and framework used throughout this thesis.

.1 Points and Vectors

A point is a vector in $\mathbb{R}^N$ Euclidean space. It also can be represented as a point in $\mathbb{R}^{N+1}$ in Homogeneous coordinates.

Homogeneous coordinates provide a natural way of representing transformations in projective geometry. Formulas involving Homogeneous coordinates are more symmetric and simpler compared to their Euclidean counterparts.

In particular a point in $3D$ is represented using a Euclidean 3-vector or a Homogeneous 4-vector with the last coordinate usually normalized to one:

$$(x, y, z) \backsimeq (x, y, z, 1) \tag{1}$$

.2 Rigid Transformations

The transformations of a robot can be represented using a rigid transformation in $3D$.

A rigid transformation preserves the distance between two points. A rigid transformation T in $3D$is defined such that when operated on a vector $\mathbf{x} \in \mathbb{R}^3$ produces:

$$T(\mathbf{x}) = \mathbf{x}' = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \mathbf{t} \tag{2}$$

where R is an orthogonal transformation and t is the translation of the origin.

The same transformation can be represented more compactly in Homogeneous coordinates as:

$$T(\mathbf{x}) = \mathbf{x}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} & R & & \mathbf{t} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3}$$

The orthogonal transformation R describes a rotation in $3D$, where $R \in SO(3)$.

.3 Optimizing Systems of Equations

In this section we give an overview of the optimization techniques by starting from the systems of linear equations and leading to the derivation of the Kalman filter. Here we give a different perspective to the problem by extending the con-

ventional least squares solution into dynamic models which leads to the Kalman filter derivation. Finally we extend that to non-linear models with particular emphasis on SLAM.

Irrespectively of the method being used, any optimization method tries to solve a linear system of equation which can be written as $A\mathbf{x} = \mathbf{b}$. In SLAM, $\{x_1, \dots x_n\} \in \mathbf{x}$ is called the state which includes the camera parameters and the landmarks. In the expanded form this equation can be written as:

$$\begin{bmatrix} a_{11}x_1 & a_{12}x_2 & \dots & a_{1(n-1)}x_{n-1} & a_{1n}x_n \\ a_{21}x_1 & a_{22}x_2 & \dots & a_{2(n-1)}x_{n-1} & a_{2n}x_n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{(m-1)1}x_1 & a_{(m-1)2}x_2 & \dots & a_{(m-1)(n-1)}x_{n-1} & a_{(m-1)n}x_n \\ a_{m1}x_1 & a_{m2}x_2 & \dots & a_{m(n-1)}x_{n-1} & a_{mn}x_n \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{(n-1)} \\ p_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{(m-1)} \\ b_m \end{bmatrix} \tag{4}$$

This can be decomposed in to a set of equations where each line corresponds to a single measurement. The least squares solution to this problem can be obtained using the pseudo inverse of A such that:

$$\mathbf{x} = \left(A^{\mathrm{T}}A\right)^{-1} A^{\mathrm{T}}\mathbf{b} \tag{5}$$

This solution, weights each measurement equally. Such a solution will not be robust enough against noise, so each measurement has to be weighted according to their importance.

If we represent the uncertainty associated with each measurement explicitly

such that:

$$E\left(\left[Ax-b\right]\left[Ax-b\right]^{\mathrm{T}}\right)=\Sigma_b \tag{6}$$

The weighted least squares equation can be obtained as:

$$\left(A^{\mathrm{T}}\Sigma_b^{-1}A\right)\mathbf{x}=A^{\mathrm{T}}\Sigma_b^{-1}\mathbf{b} \tag{7}$$

Where $\left(A^{\mathrm{T}}\Sigma_b^{-1}A\right)$ is termed the information matrix $P_{\mathbf{x}}$ of the state vector $\mathbf{x}$. The inverse of this matrix is the covariance matrix of the state ($\Sigma_x = P_{\mathbf{x}}^{-1}$).

Usually the measurements can be recursive, so the linear system has to be appended with new information. If we assume the the state to be static, with new observations a partitioned system with respective to new and old measurements can be obtained. Let the subscripts "n" represent the new measurements. Then the equation can be written as:

$$\begin{bmatrix} A \\ A_n \end{bmatrix}\begin{bmatrix} \mathbf{x} \end{bmatrix}=\begin{bmatrix} \mathbf{b} \\ \mathbf{b}_n \end{bmatrix} \tag{8}$$

If the covariances of the new and old measurements are $\Sigma_{\mathbf{b}}$ and $\Sigma_{\mathbf{b}_n}$ respectively, it can be seen from inspection that the new information matrix becomes:

$$P_{\mathbf{x}}=\begin{bmatrix} A \\ A_n \end{bmatrix}\begin{bmatrix} \Sigma_{\mathbf{b}}^{-1} & 0 \\ 0 & \Sigma_{\mathbf{b}_n}^{-1} \end{bmatrix}\begin{bmatrix} A^{\mathrm{T}} & A_n^{\mathrm{T}} \end{bmatrix} \tag{9}$$

by expanding the matrix blocks as products we get:

$$P_{\mathbf{x}} = \left[\; A\Sigma_{\mathbf{b}}^{-1}A^{\mathrm{T}} \; \right]^{-1} + \left[\; A_n\Sigma_{\mathbf{b}_n}^{-1}A_n^{\mathrm{T}} \; \right]^{-1} \tag{10}$$

The equation 10 shows that the final information matrix is the sum between the information of the system already existed and the new information gained from the measurements. Let $\tilde{\Sigma}^{-1} = A\Sigma_{\mathbf{b}}^{-1}A^{\mathrm{T}}$. By inverting this solution we can obtain the final covariance matrix of the system to be:

$$\Sigma_x = \tilde{\Sigma_x} - \left(\tilde{\Sigma}_x^{-1} + A^{\mathrm{T}}\Sigma_{b_n}^{-1}A\right)^{-1} A^{\mathrm{T}}\Sigma_{b_n}^{-1}A\tilde{\Sigma_x} \tag{11}$$

In the equation 11, the term $\left(\tilde{\Sigma}_x^{-1} + A^{\mathrm{T}}\Sigma_{b_n}^{-1}A\right)^{-1} A^{\mathrm{T}}\Sigma_{b_n}^{-1}$ is known as the gain, and is usually represented by K.

Through a similar manipulation it can be shown that, with new measurements the final state should be:

$$\mathbf{x} = \mathbf{x} + K\left(\mathbf{b}_n - A_n\tilde{\mathbf{x}}\right) \tag{12}$$

In this derivation, we have assumed that the system state is not dynamically changing. So all equations in the system are only measurements. But we can let the system state to change as well. If the system is evolving dynamically, there should be a separate set of equations other than the measurements to account for that dynamic change.

When the system is changing we can categorize the set of equations into blocks

as follows, where $\{k_0, k_1, \ldots, k-1, k\}$ are the time steps:

$$\begin{bmatrix} A_0 & 0 & \ldots & 0 & 0 \\ H_0 & -I & \ldots & 0 & 0 \\ 0 & A_1 & \ldots & 0 & 0 \\ 0 & H_1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & A_k \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_k \end{bmatrix} = \begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_k \end{bmatrix} \tag{13}$$

Here $\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k\}$ are the state vectors at each time step starting from $t = 0$ to $t = k$. Let the corresponding observation vectors to be $\{\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_k\}$. It should be noted that the equations which relates a given state with its previous state are called predictions. Though the above system is not treating them separately, the standard Kalman filter treats the measurement and the prediction updates separately.

When treated separately, the prediction equation of the Kalman filter can be written as:

$$\begin{aligned} \mathbf{x}_{t|(t-1)} &= A\left(\mathbf{x}_{(t-1)}\right) + \mathbf{u} \\ \mathbf{z}_t &= H\left[\mathbf{x}_{t|(t-1)}\right] + \mathbf{v} \end{aligned} \tag{14}$$

where $\mathbf{u}$ and $\mathbf{v}$ are the prediction and the measurement noise.

Updating the system using the measurements can be done is a similar manner as described earlier with a static state. When the system is changing dynamically, the gain K is called the Kalman gain. This gain formulation is bit different from the most commonly used Kalman gain derivation. In chapter 4 we explain the difference between these formulations and show how this difference leads to the efficient dimensionality reduction technique we present.

.3.1 The Extended Kalman Filter

In the previous section we assumed linear models. But in SLAM the prediction and the measurement models are non linear. If we let the prediction and observation models to be nonlinear (but differentiable) functions f and h respectively, the prediction and the measurement equations can be written as:

$$\begin{aligned} \mathbf{x}_{t|(t-1)} &= f\left(\mathbf{x}_{(t-1)}\right) + \mathbf{u} \\ \mathbf{z}_t &= h\left[\mathbf{x}_{t|(t-1)}\right] + \mathbf{v} \end{aligned} \tag{15}$$

To use the Kalman filter, these equations has to be linearized.

.3.1.1 Linearization

Rigid transformations are members of the Lie group $SE(3)$. The transformation $C \in SE(3)$ can be represented as:

$$R \in SO(3), \mathbf{t} \in \mathbb{R}^3 \tag{16}$$

$$C = \left[\begin{array}{ccc|c} & R & & \mathbf{t} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \in SE(3) \tag{17}$$

The lie-algebra $se(3)$ is the set of 4×4 matrices corresponding to differential translations and rotations. There are 6 generators of the algebra are:

$$G1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} G2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} G3 \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
$$G4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} G5 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} G6 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (18)$$

Then elements of $se\,(3)$ is represented as a multiple of generators:

$$\sum_{i=1}^{6} \alpha_i G_i \in se\,(3) \quad (19)$$

The exponential map from $se\,(3)$ to $SE\,(3)$ is the exponential on a linear combination of the generators. Let the the first 3 coefficients related to the translation be $\mathbf{u} = \{\alpha_1, \ldots, \alpha_3\}$ and the remaining 3 related to the rotation be $\mathbf{w} = \{\alpha_4, \ldots, \alpha_6\}$ so the mapping can be approximated as:

$$exp\left(\sum_{i=1}^{6} \alpha_i G_i\right) = I + \left[\begin{array}{c|c} \mathbf{w_x} & \mathbf{u} \\ \hline 0 & 0 \end{array}\right] + \frac{1}{2!}\left[\begin{array}{c|c} \mathbf{w_x}^2 & \mathbf{w_x u} \\ \hline 0 & 0 \end{array}\right] + \frac{1}{3!}\left[\begin{array}{c|c} \mathbf{w_x}^3 & \mathbf{w_x}^2\mathbf{u} \\ \hline 0 & 0 \end{array}\right] + \ldots \quad (20)$$

.3.2 The Jacobians

Let $C = \left[\begin{array}{ccc|c} & R & & \mathbf{t} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \in SE(3)$ and $\mathbf{x} \in \mathbb{R}^3$. Then a transformation is given by:

$$\begin{aligned} \mathbf{y} = f(C, \mathbf{x}) &= \left[\begin{array}{c|c} R & \mathbf{t} \end{array}\right] \left[\begin{array}{c} \mathbf{x} \\ 1 \end{array}\right] \\ &= R\mathbf{x} + \mathbf{t} \end{aligned} \tag{21}$$

Differentiating with respect to the vector $\mathbf{x}$ is straightforward as $\mathbf{y}$ is linear in $\mathbf{x}$:

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = R \tag{22}$$

Differentiating with respect to the generators can be done by expanding the exponential as:

$$exp\left(\sum_{i=1}^{6} \alpha_i G_i \in se(3)\right) = I + \sum_{i=1}^{6} \alpha_i G_i \in se(3) \tag{23}$$

To yield:

$$\frac{\partial \mathbf{y}}{\partial \alpha} = \left[\begin{array}{c|c|c} G1\mathbf{y} & \ldots & G6\mathbf{y} \end{array}\right] \tag{24}$$

When a landmark is observed on an image, the projection has to be applied to bring the transformed coordinate onto the image plane:

$$\mathbf{z} = T(\mathbf{y}) \tag{25}$$

The final Jacobian can be obtained using the chain-rule such that:

$$\frac{\partial \mathbf{z}}{\partial \alpha} = \frac{\partial \mathbf{z}}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \alpha} \tag{26}$$

References

P. Anuta. Spatial registration of multispectral and multitemporal digital imagery using fast fourier transform techniques. *Geoscience Electronics, IEEE Transactions on*, pages 353–368, 1970. doi: 10.1109/TGE.1970.271435. 25

A. Azarbayejani and A. Pentland. Recursive estimation of motion, structure, and focal length. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 562–575, 1995. doi: 10.1109/34.387503. 50, 53

A. Bachrach, S. Prentice, R. He, and N. Roy. Range–robust autonomous navigation in gps-denied environments. *Journal of Field Robotics*, pages 644–666, 2011. 96, 109

T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot. Consistency of the ekf-slam algorithm. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3562–3568. IEEE, 2006a. doi: 10.1109/IROS.2006.281644. 14

T. Bailey, J. Nieto, and E. Nebot. Consistency of the fastslam algorithm. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 424–429. IEEE, 2006b. doi: 10.1109/ROBOT.2006.1641748. 54

C. Berenstein, L. Kanal, D. Lavine, and E. Olson. A geometric approach to subpixel registration accuracy. *Comput. Vision Graph. Image Process.*, pages 334–360, 1987. doi: 10.1016/S0734-189X(87)80146-9. 25, 27

J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Computer Vision ECCV'92*, volume 588, pages 237–252. Springer Berlin Heidelberg, 1992. doi: 10.1007/3-540-55426-2_27. 26, 28

K. Berthold and G. Brian. Determining optical flow. *Artificial Intelligence*, 17: 185–203, 1981. 26, 28

M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An atlas framework for scalable mapping. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 1899–1906. IEEE, 2003. 53

R. Bracewell, K. Chang, A. Jha, and Y. Wang. Affine theorem for two-dimensional fourier transform. *Electronics Letters*, page 304, 1993. doi: 10.1049/el:19930207. 33

E. Brunskill and N. Roy. Slam using incremental probabilistic pca and dimensionality reduction. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 342–347. IEEE, 2005. doi: 10.1109/ROBOT.2005.1570142. 54

M. Campani and A. Verri. Motion analysis from first-order properties of optical flow. *CVGIP: Image Underst.*, 56:90–107, 1992. doi: 10.1016/1049-9660(92)90088-K. 28

A. Chiuso, P. Favaro, J. Hailin, and S. Soatto. Structure from motion causally integrated over time. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 523–535, 2002. doi: 10.1109/34.993559. 50, 53

J. Civera, A.J. Davison, and J.M.M. Montiel. Inverse depth to depth conversion for monocular slam. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2778–2783, 2007. doi: 10.1109/ROBOT.2007.363892. 23

J. Civera, O. Grasa, A. Davison, and J. Montiel. 1-point ransac for ekf-based structure from motion. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3498–3504. IEEE, 2009. 19, 68

L. Clemente, A. Davison, I. Reid, J. Neira, and J. Tardós. Mapping large loops with a single hand-held camera. In *Proceedings of Robotics: Science and Systems*. The MIT Press, 2007. 52

M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, pages 647–665, 2008. doi: 10.1177/0278364908090961. 54

A. Cunningham, M. Paluri, and F. Dellaert. Ddf-sam: Fully distributed slam using constrained factor graphs. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3025–3030. IEEE, 2010. doi: 10.1109/IROS.2010.5652875. 74, 75

J. David. Model for the extraction of image flow. *Journal of Optical Society of America, A*, pages 1455–1471, 1987. doi: 10.1364/JOSAA.4.001455. 26, 28

A. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, pages 1403–. IEEE, 2003. 53

A. Davison, I. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 1052–1067, 2007. doi: 10.1109/TPAMI.2007.1049. 12, 50, 51

F. Dellaert and M. Kaess. Square root sam: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research*, pages 1181–1203, 2006. doi: 10.1177/0278364906072768. 16, 51, 98, 105

R. Deriche, Z. Zhang, Q. Luong, and O. Faugeras. Robust recovery of the epipolar geometry for an uncalibrated stereo rig. In *Computer Vision ECCV '94*, pages 567–576. Springer Berlin Heidelberg, 1994. doi: 10.1007/3-540-57956-7_64. 19

F. Devernay and O. Faugeras. Straight lines have to be straight. *Machine vision and applications*, (1):14–24, 2001. 107

A. Donate, X. Liu, and E. Collins. Efficient path-based stereo matching with subpixel accuracy. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 41:183–195, 2011. doi: 10.1109/TSMCB.2010.2049839. 27

G. Duan and A. Robert. The importance of phase in the spectra of digital type. In *Electronic Publishing: Origination, Dissemination, and Design*, pages 47–59, 1989. 28

H. Durrant-Whyte. Uncertain geometry in robotics. *Robotics and Automation, IEEE Journal of*, pages 23–31, 1988. doi: 10.1109/56.768. 12

V.N. Dvornychenko. Bounds on (deterministic) correlation functions with application to registration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-5:206–213, 1983. doi: 10.1109/TPAMI.1983.4767373. 25

E. Eade and T. Drummond. Scalable monocular slam. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, pages 469–476. IEEE Computer Society, 2006. doi: 10.1109/CVPR.2006.263. 53, 54

E. Eade and T. Drummond. Monocular slam as a graph of coalesced observations. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. doi: 10.1109/ICCV.2007.4409098. 15, 50, 51, 54, 74, 78, 79, 84, 85, 119

J. Engel, J. Sturm, and D. Cremers. Semi-dense visual odometry for a monocular camera. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1449–1456. IEEE, 2013. doi: 10.1109/ICCV.2013.183. 18

J. Engel, T. Schps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In *Computer Vision ECCV 2014*, pages 834–849. Springer International Publishing, 2014a. doi: 10.1007/978-3-319-10605-2_54. 18

J. Engel, J. Sturm, and D. Cremers. Scale-aware navigation of a low-cost quadrocopter with a monocular camera. Elsevier, 2014b. 97, 111

R. Eustice, M. Walter, and J. Leonard. Sparse extended information filters: Insights into sparsification. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3281–3288. IEEE, 2005. doi: 10.1109/IROS.2005.1545053. 13

G. Evensen. The ensemble kalman filter, theoretical formulation and practical implementation. *Ocean Dynamics*, pages 343–367, 2003. doi: 10.1007/s10236-003-0036-9. 70, 85

B. Farrell and P. Ioannou. State estimation using a reduced-order kalman filter. *Journal of the Atmospheric Sciences*, pages 3666–3680, 2001. doi: 10.1175/1520-0469. 52

O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Computer Vision ECCV'92*, pages 563–578. Springer Berlin Heidelberg, 1992. doi: 10.1007/3-540-55426-2_61. 19

D. Fleet and A. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, pages 77–104, 1990. doi: 10.1007/BF00056772. 31

D. Fleet, A. Jepson, and M. Jenkin. Phase-based disparity measurement. *CVGIP: Image Underst.* 36

D.J. Fleet and A.D. Jepson. Stability of phase information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 1253–1268, 1993. doi: 10.1109/34.250844. 26, 28, 38

H. Foroosh, J.B. Zerubia, and M. Berthod. Extension of phase correlation to subpixel registration. *Image Processing, IEEE Transactions on*, pages 188–200, 2002. doi: 10.1109/83.988953. 25

C. Forster, M. Pizzoli, and D. Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 15–22. IEEE, 2014. 96

F. Fraundorfer, L. Heng, D. Honegger, Gim H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys. Vision-based autonomous mapping and exploration using a quadrotor mav. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4557–4564. IEEE, 2012. 96

U. Frese. Treemap: An o(log n) algorithm for indoor simultaneous localization and mapping. *Autonomous Robots*, pages 103–122, 2006. doi: 10.1007/s10514-006-9043-2. 97

D. Gabor. Theory of communication. part 1: The analysis of information. *Electrical Engineers - Part III: Radio and Communication Engineering, Journal of the Institution of*, pages 429–441, 1946. doi: 10.1049/ji-3-2.1946.0074. 30

R. Hartley. In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 580–593, 1997. doi: 10.1109/34.601246. 20

M. Hayes, J. Lim, and A. Oppenheim. Signal reconstruction from phase or magnitude. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(6): 672 – 680, dec 1980. ISSN 0096-3518. doi: 10.1109/TASSP.1980.1163463. 26, 28

T. Hsu, A. Calway, and R. Wilson. Texture analysis using the multiresolution fourier transform. In *In Scandinavian Conference on Image Analysis*, pages 823–830, 1993. 28

A. Huang, E. Olson, and D. Moore. Lcm: Lightweight communications and marshalling. In *Intelligent robots and systems (IROS), 2010 IEEE/RSJ international conference on*, pages 4057–4062. IEEE, 2010. 109

H. Jin, P. Favaro, and S. Soatto. A semi-direct approach to structure from motion. *The Visual Computer*, pages 377–394, 2003. doi: 10.1007/s00371-003-0202-6. 50, 53

M. Kaess, A. Ranganathan, and F. Dellaert. isam: Incremental smoothing and mapping. *Robotics, IEEE Transactions on*, pages 1365–1378, 2008. doi: 10.1109/TRO.2008.2006706. 51, 78

M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. pages 216–235, 2012. doi: 10.1177/0278364911430419. 16, 17, 51, 74

R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, pages 35–45, 1960. doi: 10.1115/1.3662552. 12, 54

G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007. doi: 10.1109/ISMAR.2007.4538852. 15, 16, 51, 78, 93, 107, 118

S.A. Kruger and A.D. Calway. A multiresolution frequency domain method for estimating affine motion parameters. In *Image Processing, 1996. Proceedings., International Conference on*, volume 1, pages 113 –116 vol.1, sep 1996. doi: 10.1109/ICIP.1996.559445. 26, 28, 33

T. Lee, M. Leok, and N. McClamroch. Control of complex maneuvers for a quadrotor uav using geometric methods on se (3). *arXiv preprint arXiv:1003.2005*, 2010. 109

J. Leonard and H. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Springer US, 1992. doi: 10.1007/978-1-4615-3652-9. 12

J. Leonard and H. Feder. Decoupled stochastic mapping [for mobile robot amp; auv navigation]. *Oceanic Engineering, IEEE Journal of*, pages 561–571, 2001. doi: 10.1109/48.972094. 53

H. Longuet-Higgins. Readings in computer vision: Issues, problems, principles, and paradigms. chapter A Computer Algorithm for Reconstructing a Scene from Two Projections, pages 61–62. Morgan Kaufmann Publishers Inc., 1987. 19

D. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 106

L. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–236, 1989. 27

S. Maybank, T. Huang, T. Kohonen, and M. Schroeder. *Theory of Reconstruction from Image Motion*. Springer-Verlag New York, Inc., 1992. 20

P. Maybeck. *Stochastic Models, Estimation, and Control*, volume 3. Academic press, 1982. 12

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *Eighteenth National Conference on Artificial Intelligence*, pages 593–598. American Association for Artificial Intelligence, 2002. 14, 53, 54

J. Montiel, J. Civera, and A. Davison. Unified inverse depth parametrization for monocular slam. 23

H. Nagel. On the estimation of optical flow: relations between different approaches and some new results. *Artificial Intelligence*. 26, 28

R. Newcombe, S. Lovegrove, and A. Davison. Dtam: Dense tracking and mapping in real-time. In *Proceedings of the 2011 International Conference on Computer Vision*, pages 2320–2327. IEEE Computer Society, 2011. doi: 10.1109/ICCV.2011.6126513. 51

K. Ni, D. Steedly, and F. Dellaert. Tectonic sam: Exact, out-of-core, submap-based slam. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1678–1685. IEEE, 2007. 97

David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 756–777, 2004. doi: 10.1109/TPAMI.2004.17. 20, 22

K. Ok, D. Ta, and F. Dellaert. Vistas and wall-floor intersection features - enabling autonomous flight in man-made environments. In *Workshop on Visual Control of Mobile Robots ViCoMoR*, 2012. 97

W. Paul and E. Roy. Robust regression using iteratively reweighted least-squares. *Communications in Statistics - Theory and Methods*, pages 813–827, 1977. doi: 10.1080/03610927708827533. 102

J. Philip. A non-iterative algorithm for determining all essential matrices corresponding to five point pairs. *The Photogrammetric Record*, pages 589–599, 1996. doi: 10.1111/0031-868X.00066. 20

C. Richter, A. Bry, and N. Roy. Polynomial trajectory planning for quadrotor flight. In *International Conference on Robotics and Automation*, 2013. 109

O. Robert, K. Georg, and W. David. Wide-area augmented reality using camera tracking and mapping in multiple regions. *Computer Vision and Image Understanding*, pages 854–867, 2011. doi: 10.1016/j.cviu.2011.02.007. 74

Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006. 99

D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Stereo and Multi-Baseline Vision, 2001. (SMBV 2001). Proceedings. IEEE Workshop on*, pages 131 –140, 2001. doi: 10.1109/SMBV.2001.988771. 27

S. Shen, N. Michael, and V. Kumar. Autonomous multi-floor indoor navigation with a computationally constrained mav. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 20–25, 2011. doi: 10.1109/ICRA.2011.5980357. 96

S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar. Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor. In *Robotics: Science and Systems*. MIT Press, 2013. 96

T. Shibahara, T. Aoki, H. Nakajima, and K. Kobayashi. A sub-pixel stereo correspondence technique based on 1d phase-only correlation. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, pages 221–224, 2007. doi: 10.1109/ICIP.2007.4379805. 26, 28

G. Sibley, C. Mei, I. Reid, and P. Newman. Vast-scale outdoor navigation using adaptive relative bundle adjustment. In *The International Journal of Robotics Research*, pages 958–980. SAGE Publications, 2010. doi: 10.1177/0278364910369268. 17, 75, 79, 82, 83

M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman. The new college vision and laser data set. *The International Journal of Robotics Research*, pages 595–599, 2009. doi: 10.1177/0278364909103911. 86

R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *Proceedings of the 4th International Symposium on Robotics Research*, pages 467–474. MIT Press, 1988. 11, 12

J. Sola, A. Monin, and M. Devy. Undelayed landmarks initialization for monocular slam. *Relatório técnico, Laboratoire dAnalyse et dArchitecture des Systèmes, Centre National de la Recherche Scientifique (LAAS-CNRS), Toulouse, Occitania, France*, 2008. 84

H. Strasdat, J. Montiel, and A. Davison. Real-time monocular slam: Why filter? In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2657–2664. IEEE, 2010a. doi: 10.1109/ROBOT.2010.5509636. 75, 82

H. Strasdat, J. Montiel, and A. Davison. Scale drift-aware large scale monocular slam. In *Robotics: Science and Systems*. The MIT Press, 2010b. 88

H. Strasdat, A.J. Davison, J.M.M. Montiel, and K. Konolige. Double window optimisation for constant time visual slam. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2352–2359. IEEE, 2011. doi: 10.1109/ICCV.2011.6126517. 17, 78, 119

D. Ta, K. Ok, and F. Dellaert. Vistas and parallel tracking and mapping with wall–floor features: Enabling autonomous flight in man-made environments. *Robotics and Autonomous Systems*, 2014. doi: doi:10.1016/j.robot.2014.03.010. 97

S. Taylor and T. Drummond. Binary histogrammed intensity patches for efficient and robust matching. *Int. J. Comput. Vision*. 44

S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotics Research*. 51, 54

Q. Tian and M. Huhns. Algorithms for subpixel registration. *Comput. Vision Graph. Image Process.*, pages 220–233, 1986. doi: 10.1016/0734-189X(86)90028-9. 25, 27

M. Walter, R. Eustice, and J. Leonard. Exactly sparse extended information filters for feature-based slam. *The International Journal of Robotics Research*, pages 335–359, 2007. doi: 10.1177/0278364906075026. 14, 51, 54

S. Weiss and R. Siegwart. Real-time metric state estimation for modular vision-inertial systems. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4531–4537, 2011. 111

S. Weiss, D. Scaramuzza, and R. Siegwart. Monocular-slam–based navigation for autonomous micro helicopters in gps-denied environments. 28(6):854–874, 2011. 96

S. Williams, G. Dissanayake, and H. Durrant-Whyte. An efficient approach to the simultaneous localisation and mapping problem. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, pages 406–411. IEEE, 2002. 97

S. Zhang, X. Cao, F. Zhang, and L. He. Monocular vision-based iterative pose estimation algorithm from corresponding feature points. *Science China Information Sciences*, pages 1682–1696, 2010. doi: 10.1007/s11432-010-4017-6. 22