Globally Consistent Mosaicking for Autonomous Visual Navigation

Ranjith Unnikrishnan

CMU-RI-TR-02-22

Submitted in partial fulfillment of
the requirements for the degree of
Master of Science in Robotics

The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

September 2002

© 2002 Carnegie Mellon University

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of Carnegie Mellon University.

Keywords: mosaicking, localization, mapping, cyclic networks, constrained optimization, fundamental cycles, constraint basis, non-linear state estimation, Kalman filter

Abstract

Mobile robot localization from large-scale appearance mosaics has been showing increasing promise as a low-cost, high-performance and infrastructure-free solution to vehicle guidance in man-made environments. The feasibility of this technique relies on the construction of a high-resolution mosaic of the vehicle's environment. For reliable position estimation, the mosaic must be locally distortion-free as well as globally consistent. The problem of loop closure in cyclic environments that plagues this process is one that is commonly encountered in all map-building procedures, and its solution is often computationally expensive.

This document investigates the problem of map-building with observations having low spatial and temporal persistence from sensors having a short sensory horizon. By exploiting the topology of sensor observations, the problem of mosaicking can be formulated as one of constrained optimization whose solution can be obtained efficiently even for the problem scale typical to the application of interest. A basis of the space of constraints in spatial relationships between observations can be easily extracted. Extrinsically available information in the form of survey data can be treated identically to these constraints and incorporated in the map-building process. The developed framework can also be extended to accommodate incremental construction for online implementation.

Thesis supervisor: Dr. Alonzo Kelly

Research Scientist, Robotics Institute

Acknowledgements

The work presented in this document is the fruition of innumerable enjoyable discussions with my advisor, Dr. Alonzo Kelly. His invaluable guidance, endless inspiration and broad perspective on research have taught me a lot and have greatly shaped the way I conduct my work. I have often wished that I had the benefit of working with him much earlier in my academic career.

Much credit for the software implementation surrounding this work goes to the members of the Automated Material Transport System (AMTS) project at the National Robotics Engineering Consortium, in particular Mike Happold, Bryan Nagy, Patrick Rowe and Ethan Frantz, for their timely help. Thanks also to my office mates, David Steck, Matt Troup and Surya Singh, for their creative suggestions and advice, and for lending patient ears to my questions even at odd hours of the day.

Table of Contents

Section 1: Introduction 1

Section 2: Mosaic-based Localization 2

Section 3: Implementation 4

Section 4: Issues with Mosaic Construction 6

Section 5: Prior Work 8
- 5.1 Prior mosaicking work 8
- 5.2 Map Building in Cyclic Environments 8

Section 6: Observations at an Image Overlap 10
- 6.1 Differential Relationships 10
  - Point Jacobian 10
  - Compound Point Jacobian 11
  - Sequence Jacobian 12
- 6.2 One Overlap 13
  - Absolute Pose Observer 13
  - Relative Pose Observer 14
- 6.3 Compound Pose Observer 16

Section 7: Linear Mosaicking 17
- 7.1 Total Residual of a Linear Mosaic 17
- 7.2 Smooth Distortion of a Linear Mosaic 19
- 7.3 Formulation based on the Compound Pose Observer 20

Section 8: Network Mosaicking 22
- 8.1 Loop Analysis 22

Section 9: Mosaicking an arbitrary network of images 26
- 9.1 Lockdown Graph Construction 26
- 9.2 Extraction of Fundamental Cycles 26
- 9.3 Computation of the Jacobian for Pose Residuals 26

Section 10: A Constrained Optimization Approach 29
- 10.1 Priors for Non-temporally Adjacent Overlaps 30

10.2 Residual Equations for all overlaps 31
10.3 Uncertainty Modeling 31
10.4 Loop Analysis 32
10.5 Constraint Equations 33
10.6 Formulation as a Constrained Estimation 33
10.7 Constrained Estimation as a Projection Operation 35

Section 11: Incorporating Survey Information 38

Section 12: Extension to an Incremental Mapping Algorithm 42

Section 13: Results 46

Section 14: Summary and Conclusions 51

Section 15: Future Work 52

Section 16: References 53

List of Figures

Figure 1: Normal camera mounting on vehicle 2
Figure 2: Industrial AGV 3
Figure 3: Lighting and Imaging Module 4
Figure 4: Mapping rig 4
Figure 5: Portion of a segment with one constituent image highlighted 4
Figure 6: Outline of mosaic built using only dead-reckoning estimates 6
Figure 7: Point Jacobian 10
Figure 8: Compound point Jacobian 11
Figure 9: Sequence Jacobian 12
Figure 10: One overlap 13
Figure 11: Compound pose observer 16
Figure 12: Total residual minimization in linear mosaicking 17
Figure 13: Sparse structure of feature observer derivatives 19
Figure 14: Linear mosaic distortion 19
Figure 15: One-cycled network of 4 image segments 23
Figure 16: Lockdown graph corresponding to the one-cycled network 24
Figure 17: Two-cycled network of 5 image segments 25
Figure 18: Topology graph corresponding to the two-cycled network 32
Figure 19: Two-cycled network of 5 image segments 32
Figure 20: Sparse structure of Jacobian of feature observer H and the matrix of Lagrangian derivatives 35
Figure 21: Mosaic with two surveyed fiducials and associated augmented topology graph 38
Figure 22: Mosaic with three surveyed fiducials and associated augmented topology graph 40
Figure 23: Mosaic constructed using only initial pose estimates from dead-reckoning, and mosaic after consistency enforcement 46
Figure 24: Outline of ground truth, observed and corrected trajectories 47
Figure 25: Plot of image position residual with respect to true orientations 47
Figure 26: Image orientation residual with respect to true orientation 48
Figure 27: Comparison of corrected trajectories 48
Figure 28: Comparison of image position residual between mosaics corrected with and without feature residual minimization 49
Figure 29: Comparison of image orientation residual between mosaics corrected with and without feature residual minimization 49
Figure 30: Map of indoor vehicle guidepath 50

1. Introduction

Navigating from imagery is a common technique in robotics, and a considerable amount of research interest has been directed towards using vision to provide localization and motion estimation capabilities to autonomous vehicles, both on land and in the sea. Mosaic-based localization is one such technique that is motivated by the fact that many man-made environments contain substantially flat, visually textured surfaces of persistent appearance. Images from a downward-looking camera are used to track vehicle motion over a previously obtained high-resolution mosaic of the environment floor. The problem of vehicle localization is then reduced to that of two-dimensional pose recovery through template matching. The repeatability of this technique relies largely on the prior generation of an accurate mosaic that must be both locally smooth and globally consistent to be reliable. This requirement is one that is common to all map-building procedures, and typical solutions to the associated problem are often computationally expensive and do not scale well to the large datasets typical to the application of interest.

This report examines the nature of geometric coupling that exists in the problem of map-building, and develops a scalable solution for mapping large cyclic environments using data with low spatial and temporal persistence. A method for incorporation of optional external survey data into the map-building process is also developed. Finally, an extension to a framework facilitating online implementation is also presented, along with results using simulated data as well as data from real environments.

2. Mosaic-based Localization

When cameras are used for vision-based localization, the ability to render a scene permits navigation from real-time imagery. Consider the case of a camera mounted on the undercarriage of a vehicle with its axis perpendicular to the floor immediately beneath the vehicle. The observed environment, consisting of the floor of the area navigable by the vehicle, can be represented in its photorealistic richness as an appearance model in the form of a mosaic. The constructed mosaic of the floor is rendered globally consistent and stored in persistent memory. The vehicle subsequently tracks motion over the mosaic using a visual tracker which computes camera pose. The mosaic thus serves as a *map*, linking observed features to vehicle position. We thus convert the problem of vehicle localization into a simple instance of the camera pose recovery problem of computer vision and thereby introduce a field-relevant form of vision-based localization.

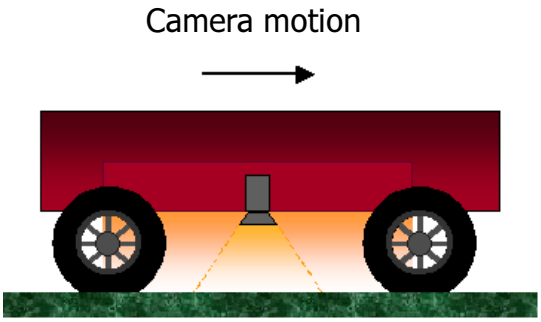


Figure 1: Simplest case of a camera mounted normal to the floor at a constant height from it. By assumption of floor planarity and mounting parameters, foreshortening of the image does not occur.

This technique differs from visual odometry in that considerable effort is expended to create a globally consistent model. It differs from landmark-based localization in that the scene is represented in an iconic form rather than as a list of landmark locations. The simplicity of the concept underlying this technique offers several advantages both in terms of implementation as well as commercial deployment:

1. The simple scene geometry simplifies processing. Since man-made floors are predominantly flat, the scene geometry is known, and tracking algorithms need not recover shape but only recover vehicle motion. The distortion of iconic features due to motion can also be predicted.
2. The use of globally consistent prior models imparts repeatability and enables higher tracking velocities. Since we are not tracking pixel regions between positions but tracking position through observed features in a persistent model, definitive locations can be extracted as long as some part of the model remains in view. If the prior model is globally consistent, reported position becomes a one-to-one function of location and the result is neither time nor path-dependent. Such a system is guaranteed to be as repeatable as the fundamental resolution of the model used.
3. No infrastructure such as wires or reflective beacons is required. This makes the system harder to sabotage and easier to install.
4. Capital cost is limited to that of a camera, lighting, off-line storage, and a capable processor - which may be required for other reasons.
5. Installation cost is limited to the labor and time required to map the environment by driving over all necessary guidepaths or areas only once.

Our application makes and exploits several simplifying assumptions:

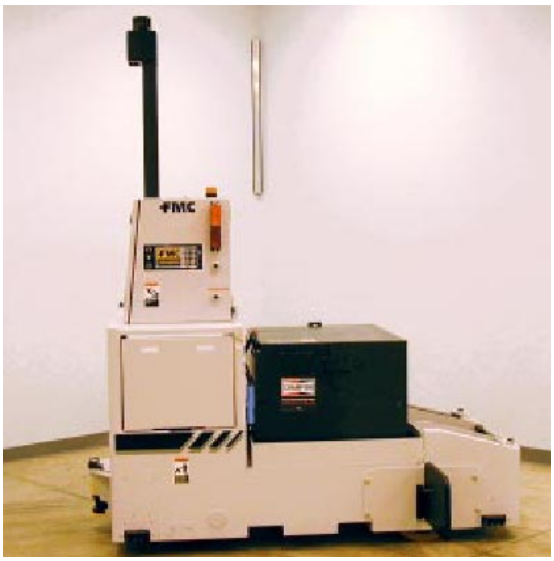


Figure 2: Industrial AGV

1. *Persistent Appearance*: While a persistent shape assumption is commonly made of scene geometry, the use of a persistently stored model of scene appearance assumes that the actual appearance of the scene will not change significantly over significant periods of time. Exceptions to this assumption are common, but the appearance change needs to be significant and extensive in order to render the present technique inoperable. It is also feasible to incorporate a learning process to accommodate slowly changing appearance.
2. *Substantially Flat 2D Scene*: While completely general 3D polygon models are certainly possible, we will assume that the scene can be represented by a mosaic mapped onto a 2D surface. While the assumption of flatness can be completely relaxed in general (e.g. in computer graphics), we will assume that the scene is flat enough that self occlusion and depth discontinuities cannot occur. These two assumptions apply, at least locally, to most man-made indoor and outdoor environments. Such environments often consist of flat surfaces punctuated by line intersections with other flat surfaces.
3. *Restricted Camera Motion*: While arbitrary camera motion is computable, we will restrict motion to be consistent with being mounted under a terrain-following vehicle. That is, the camera moves in a plane parallel to the flat scene while being oriented parallel to the terrain normal. Under these conditions, the general problem of "rendering" the scene is reduced literally to that of extracting the pixels in the rectangular region predicted to be in view.
4. *Restricted Mosaic Topology*: While not necessary in general, we will confine our attention to environments where vehicle motion is restricted to roadways or guidepaths, rather than regions wider than an image in more than one direction, except at intersections. To do so simplifies considerably the problem of constructing globally consistent mosaics. Confinement to guidepaths is required for safety reasons in commercially relevant automated guided vehicle (AGV) applications.
5. *Primary Position Estimate*: Since we will confine the application to that of vehicles, it is useful and not overly restrictive to assume the availability of an independent estimate of camera motion between frames. This primary position estimate can be used to increase reliability, tracking performance and therefore vehicle speed.

3. Implementation



Figure 3a: Lighting and Imaging Module: A custom solution to providing low-power engineered lighting and image capture facilities, that can be mounted underneath a vehicle.



Figure 3b: A custom mapping rig equipped with a downward vision module for imaging, and odometry system for providing relative motion estimates.

A custom engineered lighting and imaging module is mounted on the underside of a mapping rig, and used to capture images sequentially as the vehicle is driven over the desired guidepath. The mapping rig is equipped with two odometers on the front left and right wheels to provide differential-heading based dead-reckoning information and thus primary vehicle-image position and heading estimates from odometry. The images in each linear sequence, termed a map *segment*, are fused to obtain the appearance of a floor swath as wide as the vehicle itself. Several

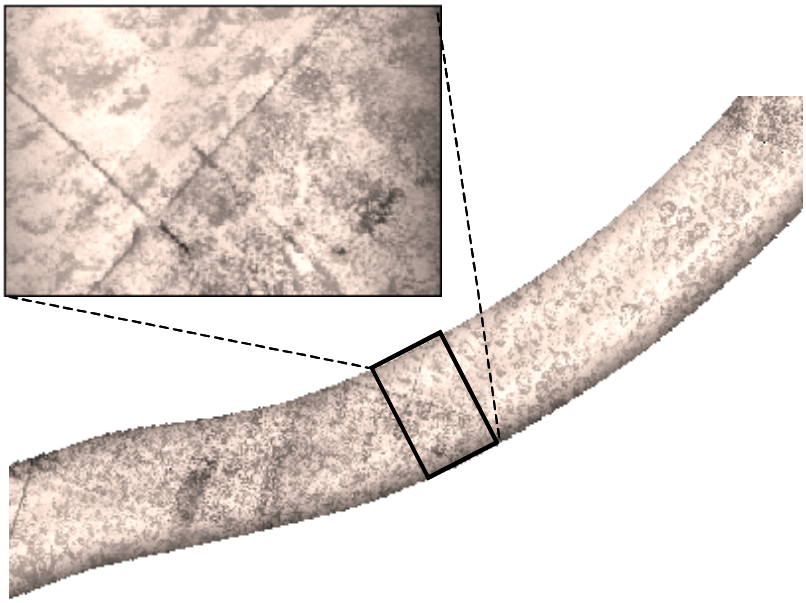


Figure 4: Portion of a segment with one constituent image highlighted

such segments in combination describe the appearance of the guidepath network within which the vehicle is designed to operate. Regions of overlap between two segments, or parts of the same segment are identified by means of fiducials, termed *lockdown points*. Such fiducials are indicative of intersection points in the guidepath where one segment “connects” to another, potentially closing a loop in the guidepath network.

Our current implementation provides a user interface using which a human operator can define or identify such fiducials to indicate potential regions of overlap within or between segments. Plausible extensions to this process include using distinct markers to serve as fiducials at intersection points in the guidepath network and automate the search for them in the captured images, or removing the need for fiducials altogether and attempting to exploit richness in floor texture to detect previously visited regions while traversing the guidepath network.

4. Issues with Mosaic Construction

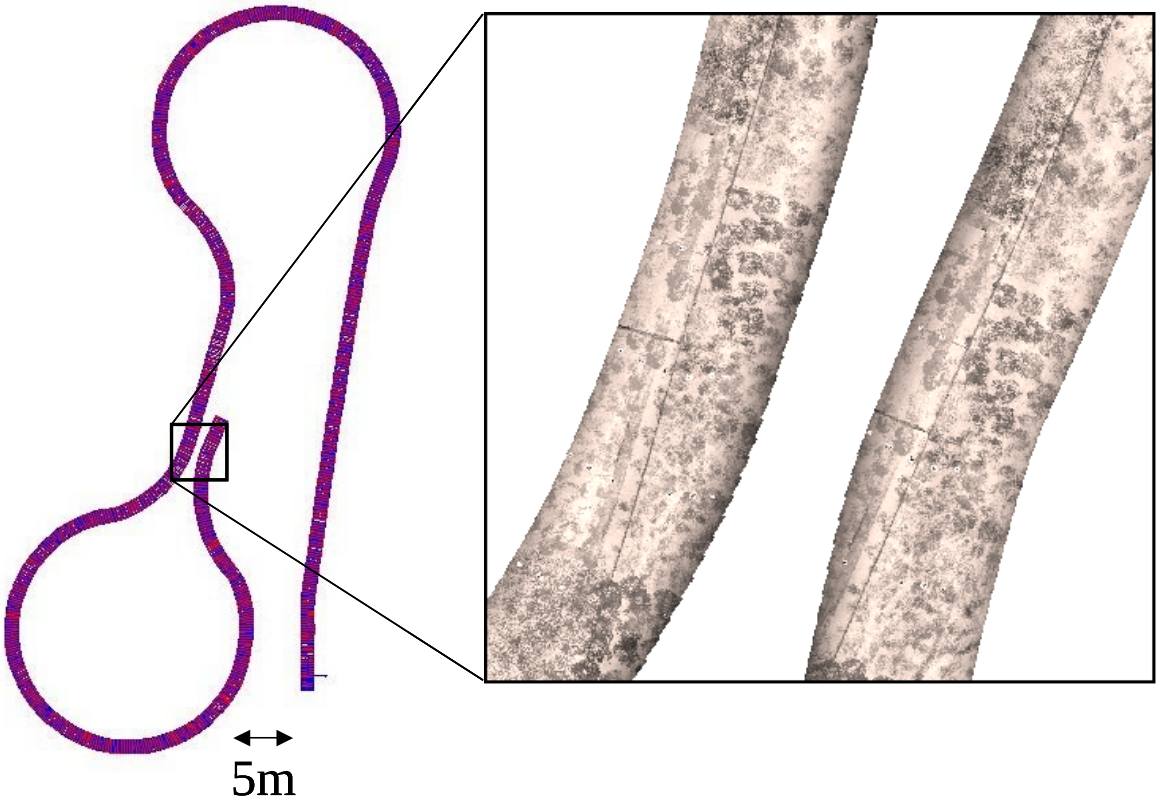


Figure 5: Outline of mosaic built using only dead-reckoning estimates. The highlighted region illustrates closure inconsistency in a portion of the trajectory that closes in on itself

Map construction using a mobile robot requires a solution to the dual problems of recovering the motion history of the robot, as well as constructing a model of its environment based on observations made during the course of the motion. For the constructed map, in our case the mosaic, to be reliable in the context of localization, it must satisfy two major conditions:

1. *Local smoothness*, whereby the registration error in adjacent images is bounded to some acceptably small value. The residual in the position of features common to two temporally adjacent images can, in principle, be reduced to a value limited only by the noise floor and finite sampling rate associated with the process of image generation. For high speed visual tracking to be possible, it is desirable to have the residual as low as possible for the features observed by the tracker to be consistent with their visual representation in the mosaic.
2. *Global consistency*, whereby the position reported at a particular location becomes neither time nor path dependent, and is uniquely represented in the constructed model. This requirement becomes apparent when the guidepath whose mosaic is to be constructed contains loops. Due to the accumulation of error in dead-reckoning estimates over large distance, a single feature present on the floor is represented in two distinct locations in the constructed mosaic. This renders the position reported by the visual tracker erroneous, as the position becomes dependent on the path taken by the vehicle and inconsistent with the true location in the reference frame of the factory.

These requirements are common to any environment model used for visual navigation, independent of the type of sensor used for the observation. The observations made by the sensor in mosaic-based localization, a calibrated downward-looking camera, differ from those of other dense sensors like laser range-finders and those used in beacon-based localization/navigation systems in that the features observed have low persistence in both spatial and temporal domains. The problem typically involves the manipulation of thousands of images for mapping loop-rich vehicle guidepaths in factories, and demands a time-efficient solution to be tractable.

5. Prior Work

5.1 Prior mosaicking work

The field of image mosaicking is a relatively old one, with no dearth of research in automated mosaicking or its applications. Several methods have been proposed, including the solution of a linear system derived from the collection of pair-wise registration matrices [3], or the frame-to-mosaic scheme [10]. Only more recently have near real-time [21] and globally-consistent solutions emerged [20]. Recently, Kang et al. [12] presented a method using a graph representation of the topology of the swaths to represent spatial and temporal adjacencies. Although the complexity for global registration is *O(mn)* in the scheme, where is *n* is the number of images and *m* is the maximum degree of a node in the topology graph, the quality of the resultant mosaic seems to depend implicitly on the proximity of each frame to its final position, and on the relatively large number of non-temporal image overlaps with respect to the number of images to be mosaicked.

5.2 Map Building in Cyclic Environments

Considerable research has been done in the field of real-time video mosaicking of the ocean-floor for navigation, exploration and wreckage visualization. Gracias and Santos-Victor [7] present several algorithms based on a projective geometry framework, and using robust matching techniques for frame-to-frame alignment. Work by Fleischer et al [6] used iterative smoother-follower techniques to reduce errors accumulated over an image chain, but no mention was made of the tractability of its extensions to networks. In the publication by Rowe and Kelly [19] related to construction of mosaics for our current application, an iterative scheme was used, whereby each segment in turn was warped to confirm to the pose and position requirements at its endpoints, in that order.

Lu and Milios [15] recognized the need for a simultaneous solution in their work on automated mapping with a laser range-finder. They distinguished spatial relationships between scan poses into those supplied by odometry and those estimated from rigorous scan matching done when revisiting areas. They then solved an overconstrained system of measurements to compute the left-pseudoinverse least-squares perturbation to absolute scan poses to enforce global consistency. The complexity of the method was however $O(n^3)$ in the number of scans, far too prohibitive for our purposes. Their scheme also had the deficiency of using hill-climbing approximations that were very sensitive to initial estimates of poses. Furthermore, their suggested incremental implementation essentially performed the same computation on all poses accumulated up to the current time instant without computational savings.

Gutmann and Konolige [9] presented a real-time autonomous mapping system producing accurate metric maps in large cyclic environments. A local registration step linked scans in a K-neighborhood of the last scan to generate a locally consistent patch, Loops were closed incrementally whenever a patch correlation scheme returned a high match score for revisited regions, with low variance and ambiguity. While the first step was of computational cost depending solely on K, the latter was essentially the same $O(n^3)$ operation of Lu and Milios, with marginal cost reductions achieved by using sparse (perhaps band) matrix techniques.

The methodology of perturbation of absolute pose is one that is common to most relevant literature that the authors have come across. Linearization of observation equations, formulating

the state variable to be estimated as a concatenation of perturbations to absolute poses, discards second order effects by definition. These effects are appreciable for problems of the scale of our mosaicking application, because the magnitude of changes in absolute poses, unlike relative ones are large. Furthermore, they contribute to violation of the fundamental assumption in linearization, that the Jacobian of the observation equations evaluated at the current (initial) estimate is approximately equal to the same evaluated at the true state value.

Work by Newman [17] on the problem of simultaneous localization and mapping (SLAM) recognized that observations of relative positional relationships between landmarks had uncertainties that were uncorrelated to uncertainties in robot position. An overcomplete, and possibly inconsistent set of such observations formed the state vector in a *relative map*, that could be updated in time linear in the number of observations. When a consistent map was explicitly required, *geometric projection filtering* was done to repeatedly project the relative map estimates onto a constraint space described by the topological relationships between the observations. The constraint equations were chosen heuristically amongst a large but finite set, such that they spanned a large subset of the observations.

In the context of our application, it is clear that high quality mosaics cannot be constructed without accounting for the coupled nature of the problem. Here, coupling means that the geometric error at one overlap is a direct function of the position of several images and an indirect function of the poses of all images in the mosaic. This document aims to formulate a model for this coupling, and analyze the memory and computational requirements associated with a tangible solution to the mosaicking problem.

A technique is presented for cost-effectively mapping cyclic network environments from low-visibility observations, that enforces global consistency while preserving existing local continuity to give usable results. The method works best in topology where the number of loops is small with respect to the number of observations, which in practice is commonly encountered and realistic in areas of limited visibility. We describe how the spanning set of constraint equations required to enforce global consistency can be easily extracted, and present an algorithm for the same. Each *loop equation* in this set corresponds to an element of the fundamental cycle basis of the topological graph describing the network of observations. On the assumption of good local estimates from odometry, an efficient scalable solution to the problem of map-building in large cyclic environments is developed for observations with a short sensory horizon. We then present a formulation that systematically relaxes the assumption of good initial estimates, while preserving the low time complexity of the previous algorithm. Our method employs the approach of total residual minimization between observations, while incorporating knowledge of prior state uncertainty and enforcing topological constraints to maintain map consistency. A framework that facilitates incremental online implementation is also proposed.

6. Observations at an Image Overlap

The process by which mosaicking takes place can be expressed as an observer process where features in overlapping images that should be in the same location are actually slightly misaligned. When this situation is expressed mathematically, it is possible to solve for the manner in which the image locations can be slightly changed in order to achieve better alignment.

6.1 Differential Relationships

This section considers the manner in which small changes in image poses affect the registration of features which are common to two or more images.

6.1.1 Point Jacobian

The "Point Jacobian" is a name for the relationship between differential output pose and the parameters of a homogeneous transformation. It is analogous to the manipulator Jacobian in robotics.

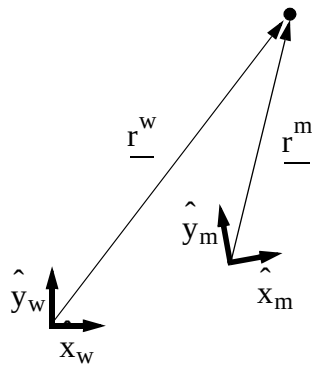


Figure 6: Point Jacobian

Consider a model frame denoted *m* and a world frame denoted *w.* The two frames involved can be any arbitrary frames. Let a point $\underline{r}^m$ be expressed in the model frame. Let the pose relating the model frame to the world frame be denoted $\underline{\rho}_m^w = \begin{bmatrix} a_m & b_m & \theta_m \end{bmatrix}^T$. We term poses referenced with respect to the world frame as *absolute poses*. Let the homogeneous transform corresponding to this absolute pose be denoted T_m^w. For the remainder of this document, the absence of an explicit superscript will imply the value is being expressed with respect to the world frame. The coordinates of the point in the world frame are given by:

$$\underline{r}^w = T_m^w \underline{r}^m$$

which can be written out as:

$$\begin{bmatrix} x^w \\ y^w \\ 1 \end{bmatrix} = \begin{bmatrix} c\theta_m & -s\theta_m & a \\ s\theta_m & c\theta_m & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^m \\ y^m \\ 1 \end{bmatrix}$$

Now the Point Jacobian is the vector partial derivative:

$$J(\underline{\rho}_m^w) = \frac{\partial}{\partial \underline{\rho}_m} \underline{r}^w = \begin{bmatrix} \frac{\partial x^w}{\partial a_m} & \frac{\partial x^w}{\partial b_m} & \frac{\partial x^w}{\partial \theta_m} \\ \frac{\partial y^w}{\partial a_m} & \frac{\partial y^w}{\partial b_m} & \frac{\partial y^w}{\partial \theta_m} \end{bmatrix} = \begin{bmatrix} 1 & 0 & (-x^m s\theta_m - y^m c\theta_m) \\ 0 & 1 & (x^m c\theta_m - y^m s\theta_m) \end{bmatrix} = \begin{bmatrix} 1 & 0 & -(y^w - b_m) \\ 0 & 1 & (x^w - a_m) \end{bmatrix}$$

This relationship gives the amount by which the world coordinates of the point change in response to small changes in the pose relating the two frames *m* and *w.* It is a central relationship in mosaicking because it quantifies the change in the position of a point caused by moving an

image a small amount.

$$\Delta \underline{r}^{w} = J(\underline{\rho}_{m}^{w})\Delta \underline{\rho}_{m}$$

6.1.2 Compound Point Jacobian

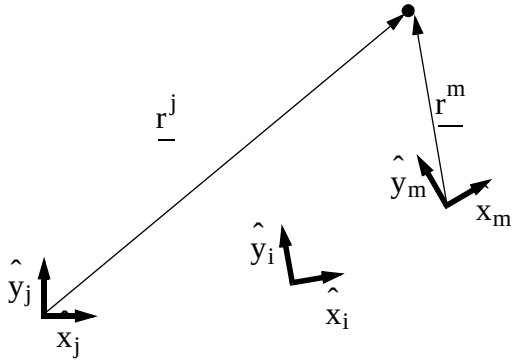


Figure 7: Compound point Jacobian

Consider now a slight variation on the above where the model frame is positioned with respect to some intermediate frame *i*, which itself is positioned with respect to another arbitrary frame *j*. Let a point $\underline{r}^{m}$ again be expressed in the model frame. Let the pose relating the model frame to the intermediate frame be denoted

$$\underline{\rho}_{m}^{i} = \left[a_{m}^{i} \; b_{m}^{i} \; \theta_{m}^{i}\right]^{T}$$

We term poses expressed with respect to frames other than the world frame as *relative poses*. Let the homogeneous transform corresponding to this relative pose relating frames *m* and *i* be denoted T_{m}^{i}, and that between frames *i* and *j* be T_{i}^{j}. Let the relative pose of the intermediate frame *i* with respect to frame *j* be expressed as .

$$\underline{\rho}_{i}^{j} = \left[a_{i}^{j} \; b_{i}^{j} \; \theta_{i}^{j}\right]^{T}$$

The coordinates of the point in frame *j* are given by:

$$\underline{r}^{j} = T_{i}^{j} T_{m}^{i} \underline{r}^{m}$$

which can be written out as:

$$\begin{bmatrix} x^{j} \\ y^{j} \\ 1 \end{bmatrix} = \begin{bmatrix} c\theta_{i}^{j} & -s\theta_{i}^{j} & a_{i}^{j} \\ s\theta_{i}^{j} & c\theta_{i}^{j} & b_{i}^{j} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta_{m}^{i} & -s\theta_{m}^{i} & a_{m}^{i} \\ s\theta_{m}^{i} & c\theta_{m}^{i} & b_{m}^{i} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^{m} \\ y^{m} \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x^{j} \\ y^{j} \\ 1 \end{bmatrix} = \begin{bmatrix} c\theta_{m}^{j} & -s\theta_{m}^{j} & c\theta_{i}^{j}a_{m}^{i} - s\theta_{i}^{j}b_{m}^{i} + a_{i}^{j} \\ s\theta_{m}^{j} & c\theta_{m}^{j} & s\theta_{i}^{j}a_{m}^{i} + c\theta_{i}^{j}b_{m}^{i} + b_{i}^{j} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^{m} \\ y^{m} \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x^{j} \\ y^{j} \\ 1 \end{bmatrix} = \begin{bmatrix} c\theta_{m}^{j}x^{m} - s\theta_{m}^{j}y^{m} + c\theta_{i}^{j}a_{m}^{i} - s\theta_{i}^{j}b_{m}^{i} + a_{i}^{j} \\ s\theta_{m}^{j}x^{m} + c\theta_{m}^{j}y^{m} + s\theta_{i}^{j}a_{m}^{i} + c\theta_{i}^{j}b_{m}^{i} + b_{i}^{j} \\ 1 \end{bmatrix}$$

where $\theta_{m}^{j} = \theta_{i}^{j} + \theta_{m}^{i}$

Now the Compound Point Jacobian is the vector partial derivative

$$J(\underline{\rho}_i^j) = \frac{\partial}{\partial \underline{\rho}_i^j}\underline{r}^k = \begin{bmatrix} \frac{\partial x^j}{\partial a_i^j} & \frac{\partial x^j}{\partial b_i^j} & \frac{\partial x^j}{\partial \theta_i^j} \\ \frac{\partial y^j}{\partial a_i^j} & \frac{\partial y^j}{\partial b_i^j} & \frac{\partial y^j}{\partial \theta_i^j} \end{bmatrix} = \begin{bmatrix} 1 & 0 & (-x^m s\theta_m^j - y^m c\theta_m^j) + (-a_m^i s\theta_i^j - b_m^i c\theta_i^j) \\ 0 & 1 & (x^m c\theta_m^j - y^m s\theta_m^j) + (a_m^i c\theta_i^j - b_m^i s\theta_i^j) \end{bmatrix}$$

$$\boxed{J(\underline{\rho}_i^j) = \frac{\partial}{\partial \underline{\rho}_i^j}\underline{r}^k = \begin{bmatrix} \frac{\partial x^j}{\partial a_i^j} & \frac{\partial x^j}{\partial b_i^j} & \frac{\partial x^j}{\partial \theta_i^j} \\ \frac{\partial y^j}{\partial a_i^j} & \frac{\partial y^j}{\partial b_i^j} & \frac{\partial y^j}{\partial \theta_i^j} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -(y^j - b_i^j) \\ 0 & 1 & x^j - a_i^j \end{bmatrix}}$$

This relationship gives the amount by which the coordinates of the point change in response to small changes in the pose relating the two frames *i* and *j*. The relationship is important in mosaicking because it quantifies the results of a small change in the relative pose of an intermediate frame.

6.1.3 Sequence Jacobian

The previous result can be used to represent how a feature moves when *every* image in a sequence is subject to small changes. We adopt a *relative pose representation* where the pose of every image is referenced with respect to its immediate predecessor in the sequence. Let each image frame be assigned a unique index *i* with numeric value increasing in sequence order. Hence the pose associated with any image of index *i* is a relative pose of the form ρ_i^{i-1}. Let the pose of the first image in the sequence be referenced with respect to the world frame. We can consider this pose of image 0 with respect to the world frame as a relative pose in this framework.

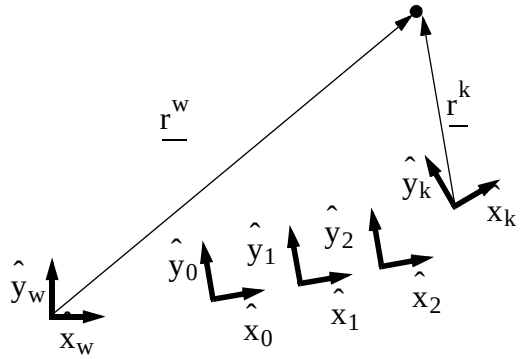


Figure 8: Sequence Jacobian

Consider a situation where a feature *f* is visible in a particular image with associated frame index *k*. The position of the feature in the world frame is given by:

$$\underline{r}_f^w = T_0^w T_1^0 \ldots T_{k-1}^{k-2} T_k^{k-1} \underline{r}^k$$

The essence of a partial derivative is to compute the effects of changes in one variable while holding any others fixed. Applying this principle here, it is clear that the total Jacobian relating changes in the feature location to changes in

each individual image pose with respect to its predecessor is:

$$\Delta \underline{r}_f^w = \begin{bmatrix} J_0 & J_1 & \dots & J_{k-1} & J_k \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \Delta a_0 & \Delta b_0 & \Delta\theta_0 \end{bmatrix}^T \\ \begin{bmatrix} \Delta a_1^0 & \Delta b_1^0 & \Delta\theta_1^0 \end{bmatrix}^T \\ \dots \\ \begin{bmatrix} \Delta a_{k-1}^{k-2} & \Delta b_{k-1}^{k-2} & \Delta\theta_{k-1}^{k-2} \end{bmatrix}^T \\ \begin{bmatrix} \Delta a_k^{k-1} & \Delta b_k^{k-1} & \Delta\theta_k^{k-1} \end{bmatrix}^T \end{bmatrix} = \tilde{J}_{0\dots k}^w \, \Delta\underline{\tilde{\Re}}_{0\dots k}$$

where $\tilde{J}_{0\dots k}^w$ is the *sequence Jacobian* and $\Delta\underline{\tilde{\Re}}_{0\dots k}$ is the sequence pose differential. The individual Jacobians are of the form:

$$J_i = R_{i-1}^w \begin{bmatrix} 1 & 0 & -(y^{i-1} - b_i^{i-1}) \\ 0 & 1 & x^{i-1} - a_i^{i-1} \end{bmatrix}$$

where the rotation matrix

$$R_{i-1}^w = \begin{bmatrix} c\theta_{i-1}^w & -s\theta_{i-1}^w \\ s\theta_{i-1}^w & c\theta_{i-1}^w \end{bmatrix}$$

with $\theta_{i-1}^w = \theta_0^w + \theta_1^0 + \dots + \theta_{i-1}^i$ may be interpreted as converting the Jacobian of the form seen in Section 6.1.2 on page 11 to the reference of frame *w.*

(a_i^{i-1}, b_i^{i-1}) is the position of the origin of frame *i* with respect to frame *i-1*, and (x^{i-1}, y^{i-1}) is the position of the feature with respect to the frame *i-1*.

6.2 One Overlap

Consider a pair of overlapping images denoted *i* and *j*. Each image has its own "model" frame position with respect to the world frame.

Let a feature in the region of overlap be denoted *m*. Unless these images are perfectly registered, each will have a different estimate of where the feature is:

$$\underline{r}_m^w(\underline{\rho}_i) = T_i^w(\underline{\rho}_i)\underline{r}_m^i \qquad \underline{r}_m^w(\underline{\rho}_j) = T_j^w(\underline{\rho}_j)\underline{r}_m^j$$

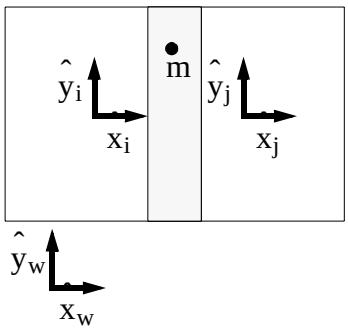


Figure 9: One overlap

6.2.1 Absolute Pose Observer

Hence, the residual z_m, or difference between them is:

$$\underline{z}_m = \underline{r}_m^w(\underline{\rho}_i) - \underline{r}_m^w(\underline{\rho}_j) = T_i^w(\underline{\rho}_i)\underline{r}_m^i - T_j^w(\underline{\rho}_j)\underline{r}_m^j$$

This is of the form of the standard observer $z = h(\underline{x})$ if we generate a state vector by adjoining

the poses of the two images into a single state vector thus:

$$\underline{x} = \begin{bmatrix} \underline{\rho}_i^T & \underline{\rho}_j^T \end{bmatrix}^T$$

Such a formulation enables computing a relationship between the residual and differential changes in either image absolute pose.

Let J_i^m and J_j^m denote the point Jacobians of the feature *m* with respect to each image pose respectively. Then on differentiating the above expression, the *change in the residual* can be written as a matrix times the *change in the adjoined state*:

$$\Delta \underline{z}_m^w = J_i^m \Delta \underline{\rho}_i - J_j^m \Delta \underline{\rho}_j = H \Delta \underline{x} = \begin{bmatrix} J_i^m & -J_j^m \end{bmatrix} \begin{bmatrix} \Delta \underline{\rho}_i \\ \Delta \underline{\rho}_j \end{bmatrix}$$

where the matrices involved are partitioned and have been written to expose their component matrices.

This final equation can be written for a number of features in an overlap region. It is clear that a minimum number of 2 unique feature points are need to over-constrain the pose of one of the images with respect to the other, since any one feature point provides two constraining equations in its (x,y) position. Provided enough unique features are used, the system of equations can be solved to minimize the residual and register the images together as close to perfectly as is possible. To do this, simply write one equation for each feature and gather all equations together into a large matrix equation:

$$\begin{bmatrix} \Delta x_1 \\ \Delta y_1 \\ \Delta x_2 \\ \Delta y_2 \\ \dots \end{bmatrix} = \begin{bmatrix} J_i^1 & -J_j^1 \\ J_i^2 & -J_j^2 \\ \dots & \dots \end{bmatrix} \begin{bmatrix} \Delta \underline{\rho}_i \\ \Delta \underline{\rho}_j \end{bmatrix}$$

The notation J_i^m means the point Jacobian relating changes in the position of feature *m* to the associated changes in the pose of image *i*. The feature locations are distinct so:

$$J_i^1 \neq J_i^2$$

The image poses are also distinct so:

$$J_i^m \neq J_j^m$$

By keeping the absolute pose of one of the images as a reference, the absolute pose of other can be determined, in a least-squares sense, by solving the system of overconstrained equations.

The advantage of this formulation is that changes in the absolute poses of images are available directly - they do not need to be computed from changes in relative poses as in the next section.

6.2.2 Relative Pose Observer

The residual can be expressed to depend only on the relative pose of the two images in an overlap.

Recall from above that the residual, z_m, or difference between feature positions is:

$$\underline{z_m} = \underline{r_m^w}(\underline{\rho}_i) - \underline{r_m^w}(\underline{\rho}_j) = T_i^w(\underline{\rho}_i)\underline{r_m^i} - T_j^w(\underline{\rho}_j)\underline{r_m^j}$$

We can premultiply this expression by T_w^i to express the result in the image *i* model frame:

$$\underline{z_m^i} = T_w^i \underline{z_m^w} = \underline{r_m^i} - T_j^i(\underline{\rho}_j^i)\underline{r_m^j}$$

This is also of the form of the standard observer $z = h(\underline{x})$ if the state vector is the relative pose of the two images:

$$\underline{x} = \underline{\rho}_j^i$$

Such a formulation also enables computing a relationship between the residual and differential changes in the relative image pose.

Note that there is now only one image pose involved - the relative pose relating *j* to *i*. Let image frame *i* function as the world frame and image pose *j* function as the model frame. Let J_j^i denote the point Jacobian of the feature with respect to the pose of image *j* relative to image *i*. On differentiating the above expression, the *change in residual* can be written as a matrix times the *change in state*:

$$\Delta\underline{z_m^i} = J_j^i \Delta\underline{\rho}_j^i = H\Delta\underline{x}$$

This final equation can be written for a number of features in an overlap region. Provided enough unique features are used, the system of equations can be solved as before to minimize the residual and register the images together as close to perfectly as is possible. To do this, simply write one equation for each feature and gather all equations together into a large matrix equation:

$$\begin{bmatrix} \Delta x_1 \\ \Delta y_1 \\ \Delta x_2 \\ \Delta y_2 \\ \cdots \end{bmatrix} = \begin{bmatrix} {}_1J_j^i \\ {}_2J_j^i \\ \cdots \end{bmatrix} \begin{bmatrix} \Delta\underline{\rho}_j^i \end{bmatrix}$$

The notation ${}_2J_j^i$ means the point Jacobian relating changes in the position of feature 2 (expressed in the frame of image *i*) to the associated changes in the pose of image *j* relative to image *i*. The feature locations are distinct so:

$${}_1J_j^i \neq {}_2J_j^i$$

This approach requires an ordering of poses in that each image, except from one distinguished image, will be positioned relative to some “earlier” image. Two images may share the same predecessor but no image has more than one predecessor. Under these conditions, a relative pose can be defined for every image except the root of the predecessor tree. Reversion to an absolute pose representation can be done by simply assigning an absolute pose to an arbitrarily chosen base image, and then sequentially computing absolute poses for each neighboring image.

6.3 Compound Pose Observer

Consider now a situation where two sequences of images, each comprising a kinematic chain, are known to overlap at a given feature. Let each sequence have a distinguished frame (*p* and *q* in the figure) associated with it whose pose is known with respect to the world. This may be a particular image frame or any other frame which is affixed to the sequence. Let a feature *m* be visible in image *i* of the first sequence and image *j* of the second sequence. Ultimately all image poses will be placed in a single vector so their indices must be distinct.

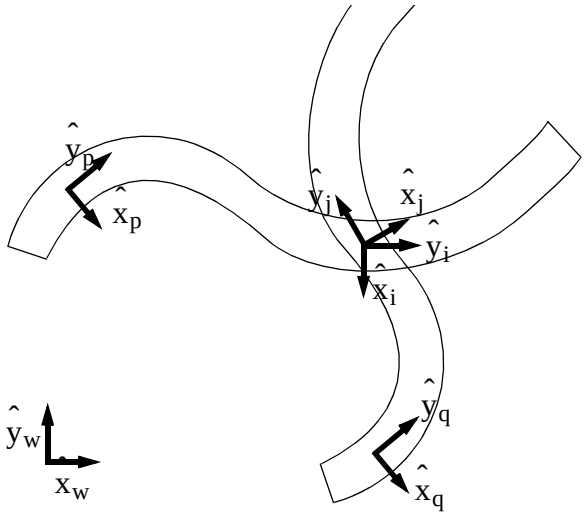


Figure 10: Compound pose observer

The pose of the feature *m* can be computed from each sequence and they are likely to disagree:

$$\underline{r}_m^w(\underline{\tilde{\Re}}_{p\ldots i}) = T_p^w T_i^p(\underline{\tilde{\Re}}_{p\ldots i})\underline{r}_m^i$$

$$\underline{r}_m^w(\underline{\tilde{\Re}}_{q\ldots j}) = T_q^w T_j^q(\underline{\tilde{\Re}}_{q\ldots j})\underline{r}_m^j$$

where $\underline{\tilde{\Re}}_{p\ldots i}$ is the vector of relative poses of all images from *i* to *p+1* and T_i^p is the compound homogeneous transform relating frame *i* to frame *p*. We can consider the pose of the image *p* relative to the world as its "relative" pose in this framework.

The residual, z_m, or difference between them is:

$$\underline{z}_m = \underline{r}_m^w(\underline{\tilde{\Re}}_{p\ldots i}) - \underline{r}_m^w(\underline{\tilde{\Re}}_{q\ldots j}) = T_i^w(\underline{\tilde{\Re}}_{p\ldots i})\underline{r}_m^i - T_j^w(\underline{\tilde{\Re}}_{q\ldots j})\underline{r}_m^j$$

This is of the form of the standard observer $z = h(\underline{x})$ if we generate a state vector by adjoining the poses of the two images into a single state vector thus:

$$\underline{x} = \begin{bmatrix} \underline{\tilde{\Re}}_{p\ldots i} \\ \underline{\tilde{\Re}}_{q\ldots j} \end{bmatrix}$$

Such a formulation enables computing a relationship between the residual and differential changes in any image relative pose in either sequence.

Lets differentiate the above expression. Let $\tilde{J}^w_{p\ldots i}$ and $\tilde{J}^w_{q\ldots j}$ denote the sequence Jacobians of the feature with respect to each image sequence respectively. Then, the *change in the residual* can be written as a matrix times the *change in the adjoined state*:

$$\Delta\underline{z}_m^w = \tilde{J}^w_{p\ldots i}\Delta\underline{\tilde{\Re}}_{p\ldots i} - \tilde{J}^w_{q\ldots j}\Delta\underline{\tilde{\Re}}_{q\ldots j} = H\Delta\underline{x} = \begin{bmatrix} \tilde{J}^w_{p\ldots i} & -\tilde{J}^w_{q\ldots j} \end{bmatrix} \begin{bmatrix} \Delta\underline{\tilde{\Re}}_{p\ldots i} \\ \Delta\underline{\tilde{\Re}}_{q\ldots j} \end{bmatrix}$$

where the matrices involved are partitioned and have been written to expose their component matrices.

7. Linear Mosaicking

Linear mosaics are the simplest mosaic to construct. A smooth linear mosaic can be constructed simply by sequentially minimizing the feature position residual for each overlap region in the mosaic as it is traversed from start to finish. This problem is solvable in insignificant time because the unknowns are not coupled at all. The problem can hence be partitioned into smaller independent subproblems of number equal to the number of images involved. The linear mosaic, or segment, can then be termed as locally (or internally) smooth, whereby the maximum distortion error due to incorrect pose fits is bounded to an arbitrarily small value.

In practise, a reasonably well-calibrated differential-heading odometer provides accurate estimates of relative pose over short distances. On the platforms used in our project, this error has been found to reflect on distortion that is less than a pixel between successive images. The problem of interest, however, is that of the correction of this error on its appreciable accumulation over large distances.

7.1 Total Residual of a Linear Mosaic

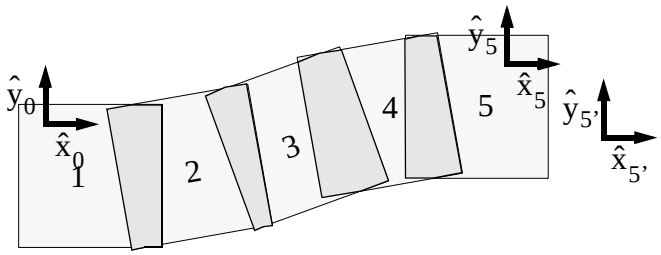


Figure 11: Total residual minimization in linear mosaicking

A slightly more difficult problem than linear mosaicking is the problem of distorting a linear mosaic so that its endpoint is moved a small amount in position and orientation. Consider the situation outlined in figure (11) where a linear sequence of images overlap such each image contains a portion of the scene in common with both the image before and the image after it in the sequence. Let each image be assigned a unique index, and let each image pose be expressed with respect to the world frame in an *absolute pose representation*.

Suppose it is necessary to move image frame 5 to pose 5' in a manner which minimizes the overall residual of the features at each image overlap to the existing smooth linear mosaic. This can be accomplished with the left pseudoinverse. Let at least two features be identified and their new locations be used to constrain the pose of frame 5' relative to frame 5. For each feature in each overlap, we can express the change in its residual as a function of its Jacobian, taken with respect to its absolute pose, and the change in the adjoined pose vector. We define overlap *i* to be that between images *i* and *i+1*. Let z_m^i refer to the residual of the *m*-th feature in the *i*-th overlap, and ${}_{i+1}J_m^i$ refer to the Jacobian relating the change in world coordinates of the *m*-th feature in the *i*-th overlap, with changes in the absolute pose of image *i+1*. Then the equation for each overlap feature can be written as:

$$\Delta z_m^i = {}_{i+1}J_m^i \Delta\rho_{i+1}^w - {}_iJ_m^i \Delta\rho_i^w = \begin{bmatrix} -{}_iJ_m^i & {}_{i+1}J_m^i \end{bmatrix} \begin{bmatrix} \Delta\rho_i^w \\ \Delta\rho_{i+1}^w \end{bmatrix}$$

For the features relating the frames 5 and 5', the residual of each feature is given by the difference of its position and a desired world position. Alternately the features can be thought of

as lying in an overlap with $i = 5$. Then, the change in their residuals can be related as:

$$\Delta z_m^5 = -{}_5J_5^m \Delta\rho_5^w = \left[{}_5J_5^m\right]\left[\Delta\rho_5^w\right]$$

Concatenating the equations for all features in a given overlap gives:

$$\begin{bmatrix} \Delta z_0^i \\ \Delta z_1^i \\ \dots \\ \Delta z_{m_i}^i \end{bmatrix} = \begin{bmatrix} 0 & \dots & -{}_iJ_0^i & {}_{i+1}J_0^i & \dots & 0 \\ 0 & \dots & -{}_iJ_1^i & {}_{i+1}J_1^i & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & -{}_iJ_m^i & {}_{i+1}J_m^i & \dots & 0 \end{bmatrix} \begin{bmatrix} \Delta\rho_0^w \\ \dots \\ \Delta\rho_i^w \\ \Delta\rho_{i+1}^w \\ \dots \\ \Delta\rho_5^w \end{bmatrix}$$

where m_i refers to the number of feature points in the *i*-th overlap. Writing such a set of equations for all overlaps results in a system of the form:

$$\Delta\underline{z} = H\Delta\underline{x}$$

where $\Delta\underline{x}$ is a column vector of adjoined absolute pose differentials, and $\Delta\underline{z}$ is a column vector of adjoined changes in feature residuals.

The equations are overdetermined, and will therefore be solvable by the left pseudoinverse:

$$\Delta\underline{x} = [H^T H]^{-1} H^T \Delta\underline{z}$$

If the pose of the first image is kept fixed to its initial value, and that of the last image is set to be equal to its desired value, the first and last columns of J's reduce to zero and can be removed from the matrix altogether. This leaves the H matrix to be of size $2M \times 3(N-2)$, where M is the total number of feature points, and N is the number of images.

Memory Usage and Computational complexity

It may be observed that for a linear mosaic of n_i images with n_f features per overlap, the H matrix is of size $2n_f n_i \times 3(n_i - 1)$ and contains approximately $n_i \times 12n_f = 12n_f n_i$ non-zero elements.

With careful implementation, the Hessian $H^T H$ is computable with $O(n_f n_i)$ cost. For linear mosaics, $H^T H$ is of a block tridiagonal form with each block of size 3-by-3. Positive definite matrices of this form can be efficiently inverted through Cholesky factorization in time sub-cubic in n_i.

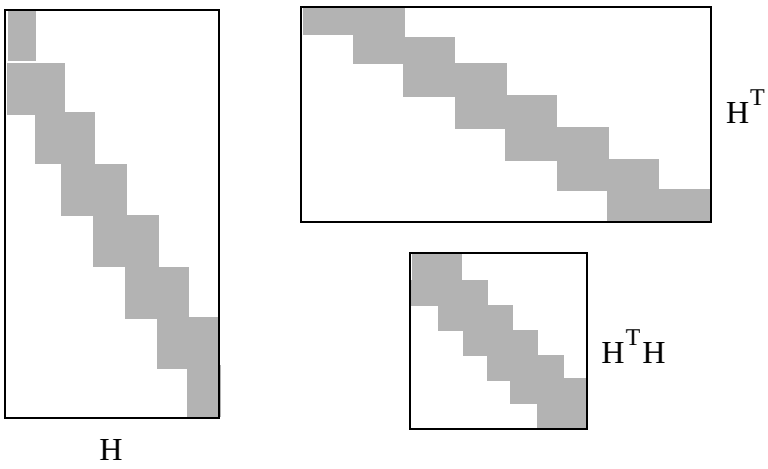


Figure 12: Sparse structure of feature observer derivatives. Grayed regions correspond to non-zero elements. Note the block-tridiagonal structure of the Hessian approximation $H^T H$

However a solution of this form does not scale well to a network of images. Overlaps between non-temporally adjacent images introduce off-diagonal block elements in the Hessian matrix. These entries are crucial to the generation of a globally consistent mosaic, but destroy the block-diagonal structure of the $H^T H$ matrix making inversion expensive. This in turn makes the problem of mosaicking several thousands of networked images in this manner computationally intensive.

The next subsection attempts to formulate the problem in a manner that utilizes and preserves the continuity of temporally adjacent images to reduce the dimensionality of the problem, while enforcing consistency at the end-points.

7.2 Smooth Distortion of a Linear Mosaic

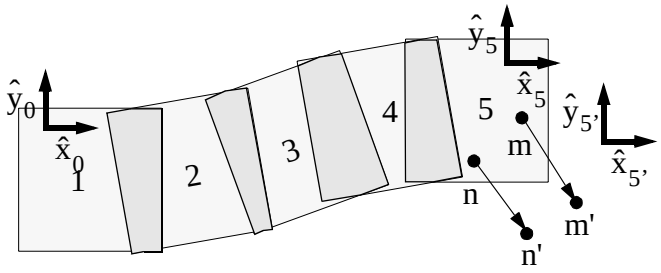


Figure 13: Linear mosaic distortion

Suppose it is necessary to move image frame 5 to pose 5' in a manner which causes minimal overall distortion to an existing smooth linear mosaic. We adopt a *relative pose representation*, where the pose of each image in the sequence is expressed with respect to its immediate predecessor in the sequence.

Let two features, *m* and *n*, be identified and their new locations be used to constrain the pose of frame 5' relative to frame 5. For feature *m*, we can use the sequence Jacobian from frame 1 to frame 5 to constrain the *position* of frame 5 as:

$$\Delta \underline{z}_m^1 = \tilde{J}_{1\ldots5}^1 \Delta \underline{\tilde{\Re}}_{1\ldots5}$$

The second feature point, *n*, can be interpreted as constraining the *pose* of frame 5. The change in

orientation residual can be expressed as:

$$\Delta\theta_n^1 = \Delta\theta_2^1 + \Delta\theta_3^2 + \Delta\theta_4^3 + \Delta\theta_5^4 + \Delta\theta_n^5$$

where the terms $\Delta\theta_{i+1}^i$ on the right-hand side represent the change in relative orientation between successive frames, and the term $\Delta\theta_n^5$ represents the change in relative orientation of the observed feature point in frame 5', which is zero since frame 5' is fixed.

The adjoined equations can be represented as:

$$\begin{bmatrix} \begin{bmatrix} \Delta x_m^1 \\ \Delta y_m^1 \end{bmatrix} \\ \Delta\theta_n^1 \end{bmatrix} = \begin{bmatrix} {}_mJ_1^2 & \cdots & {}_mJ_{i-1}^i & \cdots & {}_mJ_4^5 \\ \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} & \cdots & \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} & \cdots & \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \Delta\tilde{\underline{\Re}}_{1\ldots5} \end{bmatrix}$$

The left subscript on the Jacobians is used to denote the fact that the feature coordinates used in the Jacobian come from the feature whose index appears.

This set of equations is of the form:

$$\Delta\underline{z} = H\Delta\underline{x}$$

The equations are underdetermined, and will therefore be solvable by the right pseudoinverse:

$$\Delta x = H^T[HH^T]^{-1}\Delta z$$

The resulting value of Δx gives the *minimum norm perturbation* to the relative poses of the images that is required to achieve the shift of frame 5 to the desired end frame position of 5'. In this formulation, it is important that the number of equations, or the number of rows of H be at most 3, as any more would result in overconstraining the pose of frame 5, whereby the matrix H would lose row rank.

Memory Usage and Computational complexity

It may be observed that for a linear mosaic of n_i images, the approximate size of the H matrix is only $3 \times 3n_i = 9n_i$ elements, which is far less than that required by the previous subsection.

The cost of computing HH^T is only $O(n_i)$, and being of size 3-by-3 can be inverted in constant time. Hence the order of the whole process is a comfortable $O(n_i)$.

7.3 Formulation based on the Compound Pose Observer

An important case is where a large number of linear mosaics overlap at discrete intersection points. If the linear mosaics are already smooth, the right pseudoinverse again provides an efficient solution to the mosaicking problem. For this problem, we:

- enumerate all "intersections" in the mosaic
- for each intersection identify the two segments which have images participating in the intersection
- for each segment, identify the image participating in the intersection and another *key* image as far as possible from it without crossing another intersection, whose pose relative to the world is known. The coordinate frame associated with the key image is termed the *key frame*.

- for the image participating in the intersection, locate at least two features which can also be located in an image of the other segment.

When all of this is done, we can write compound pose observers for each feature by projecting the observed feature position onto all of the relative poses of both image sequences. The equations for one feature look like:

$$\Delta \underline{z}_{\underline{m}}^{w} = \tilde{J}_{p\ldots i}^{w} \Delta \underline{\tilde{\Re}}_{p\ldots i} - \tilde{J}_{q\ldots j}^{w} \Delta \underline{\tilde{\Re}}_{q\ldots j} = H \Delta \underline{x} = \begin{bmatrix} \tilde{J}_{p\ldots i}^{w} & -\tilde{J}_{q\ldots j}^{w} \end{bmatrix} \begin{bmatrix} \Delta \underline{\tilde{\Re}}_{p\ldots i} \\ \Delta \underline{\tilde{\Re}}_{q\ldots j} \end{bmatrix}$$

Multiple equations are written by respecting the unique index allocation of all images in the network and placing the individual sequence Jacobians (with the correct signs) in the correct columns of the overall coefficient matrix. It is also necessary to shift the base frame with respect to which a feature residual is expressed, to one of the images which contain that feature, in order to correctly represent the residual vector to be minimized.

This set of equations obtained will again be of the form:

$$\Delta \underline{z} = H \Delta \underline{x}$$

The equations are underdetermined, and will therefore be solvable by the right pseudoinverse:

$$\Delta x = H^{T} [H H^{T}]^{-1} \Delta z$$

If the pose of the world frame is left out of the state vector $\Delta \underline{x}$, it should not be necessary to strike an image pose to preserve invertibility of the pseudoinverse.

A major disadvantage of this approach is that the key frame of each segment is required to be sufficiently proximal to its expected final pose, as the formulation grants it only the same level of freedom of movement with respect to the world frame as that given to any other image whose pose is expressed with respect to its predecessor. This can be rectified using a weighted right pseudo-inverse formulation, with which the degree of motion of the key frames with respect to that of other images can be arbitrarily set by choosing appropriate weights. This, however, makes the solution susceptible to inaccuracies in the supplied weight values. Solutions obtained for differing weights, although satisfying the required constraints in feature residuals, will not be the same. Hence a more general scheme for building mosaics with multiple overlapping sequences is required.

8. Network Mosaicking

8.1 Loop Analysis

The problem of generating a consistent mosaic of a network of internally smooth linear segments can be solved by exploiting the topology of the network to generate end-pose constraints at the intersections. By representing the poses of all images relative to their predecessor in a segment, we convert their representation into one that can be used to preserve local continuity. By adopting any one image as our base, and fixing its frame (or pose), we essentially fix the pose of all other images that participate in an intersection with it. Hence, by representing the pose of these participating images in terms of the relative poses of all images, as encountered on traversing any kinematic path from the base to the overlap, we can solve for minimal changes in those relative poses for a supplied change in end pose, or end pose thereof.

Example 1: One-cycle topology

Consider the simple single-cycled network shown in figure (14). The images pairs (*h*,*a*), (*b*,*c*), (*d*,*e*) and (*f*,*g*) form four overlaps. Each image in a pair is registered to its counterpart i.e. the pose of one image relative to the other in each pair is known either through manual specification (by specifying the position and orientation of a common feature termed a *lockdown point*), or through a prior stage of automatically checking for and registering every pair of images that are in sufficient proximity of each other to constitute a potential overlap. Each lockdown point is associated with its own coordinate frame, termed a *lockdown frame*. In the example, the frames are assigned labels L1 through L4.

In our analysis, we shall refer to each overlap as containing a lockdown point. This does not lead to any loss of generality, as the case of registration of two images to determine the pose of one with respect to the other can be interpreted as having determined a common lockdown point in the two images, whose observed poses with respect to each of the two image frames are related by the previously determined relative pose between the two image frames.

Let us assume for now that image indices corresponding to the letters *a*, *b*, *c* and *d* are in order of increasing magnitude. As per the convention we will adopt, every image pose is referenced with respect to its immediate predecessor having a smaller image index. We also adopt the convention that homogenous transforms represented by lower-case t refer to constant quantities, or relationships between images that are determined a priori by the registration process described in the earlier paragraph. Transforms represented by upper-case T refer to unknowns which are to be determined.

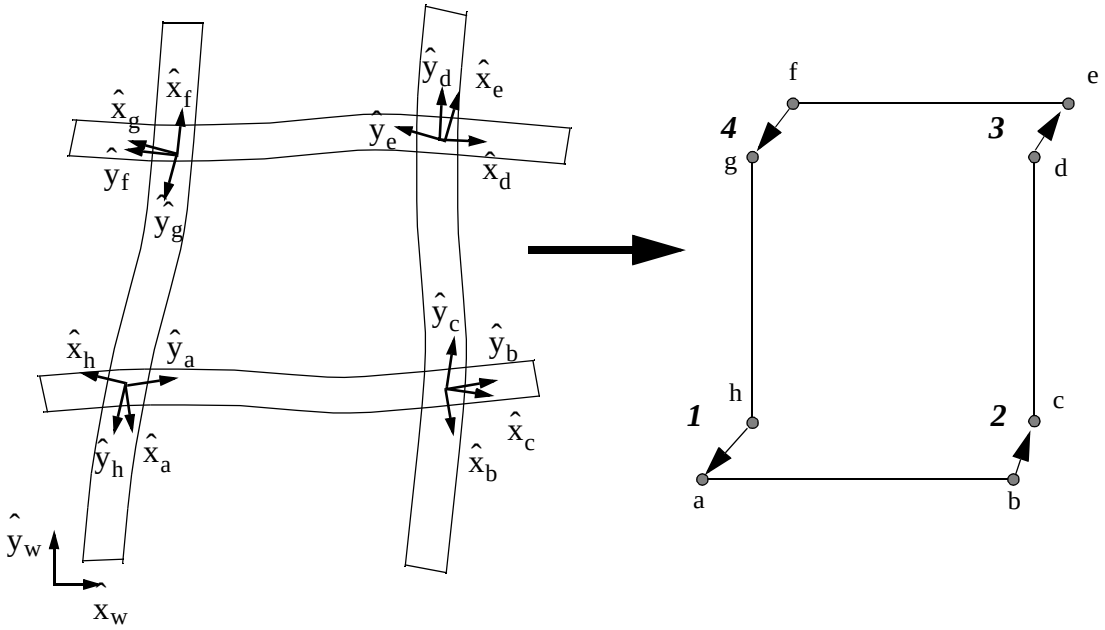


Figure 14: One-cycled network of 4 image segments

Hence the homogenous transform relating frame *b* to *a* can be written as:

$$T_b^a = T_{a+1}^a T_{a+2}^{a+1} T_{a+3}^{a+2} \ldots T_b^{b-1}$$

and similarly for *d* to *c*, etc. We refer to such an expression as a *transform chain* to describe the sequence of homogenous SE(2) transformation matrices that equate to the pose relationship on the left hand side of the equation. It is clear that each transformation matrix in the sequence is an explicit function of the relative pose described by it.

Figure (15) shows the **lockdown graph** (LG) corresponding to the above network. We define a lockdown graph as a collection of a vertex set υ, and an edge set ε. Each vertex $v \in \upsilon$ is composed of a lockdown frame L_v unique to each lockdown point, and a set of subnodes Φ. Each subnode $n \in \Phi$ belonging to a vertex v represents a unique "footprint" of the lockdown frame associated with v, i.e. there exists a subnode corresponding to each image of the overlap set that contains the lockdown point. Edges between sub-nodes correspond to pose relationships between the associated images as observed through a part of an image sequence. The LG of a set of image observations thus describes the topology of the geometric relationships within the dataset as determined by lockdown points common to overlapping non-temporally adjacent images.

From figure (15), it is clear that the lockdown graph corresponding to the example network contains one cycle. By traversing the kinematic chain corresponding to this cycle, we can represent the frame of any image with respect to any arbitrarily chosen base frame in two distinct ways, corresponding to the two distinct image sequence paths that can be taken to that image starting from the image containing the base frame. It is clear that the homogenous transform expressing the base frame relative to itself, as determined by the transform chain corresponding to the cycle, should equal identity.

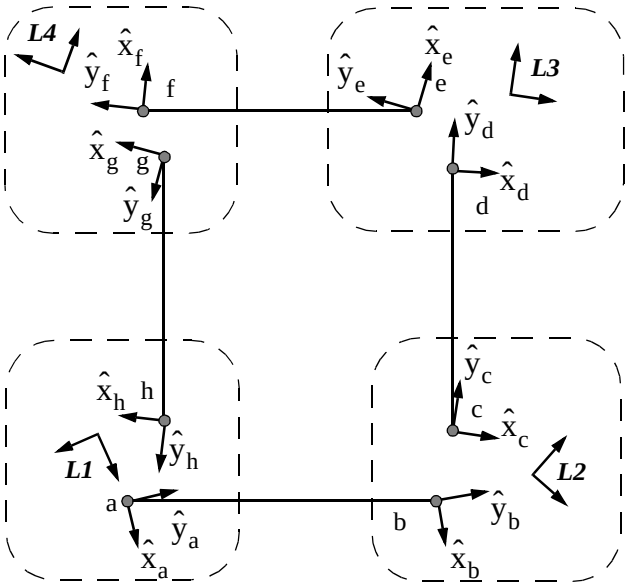


Figure 15: Lockdown graph corresponding to the one-cycled network

Mathematically,

$$T_a^a = T_b^a \cdot t_c^b \cdot T_d^c \cdot t_e^d \cdot T_f^e \cdot t_g^f \cdot T_h^g \cdot t_a^h = I_{3\times 3}$$

or

$$T_a^a = T_b^a \cdot t_{L_2}^b \cdot t_c^{L_2} \cdot T_d^c \cdot t_{L_3}^d \cdot t_e^{L_3} \cdot T_f^e \cdot t_{L_4}^f \cdot t_g^{L_4} \cdot T_h^g \cdot t_{L_1}^h \cdot t_e^{L_1} = I_{3\times 3}$$

where the 't' elements, as defined earlier, are knowns determined from the lockdown points from the four overlaps.

Taking the derivative of the above equation with respect to all the relative pose variables yields 3 constraining differential equations that comprise a system of the form

$$\Delta \underline{z} = H \Delta \underline{x}$$

where $\Delta \underline{x}$ is the augmented vector of change in relative poses of all the images in the network, and $\Delta \underline{z}$ is the observed pose residual.

This system is underconstrained and can be solved by the right pseudo-inverse as

$$\Delta x = H^T [H H^T]^{-1} \Delta z$$

Example 2: Two-cycle topology

Consider Figure (16) where an additional segment has been incorporated so as to generate another fundamental cycle in the lockdown graph. This cycle addition generates another equation of the form of that in example 1.

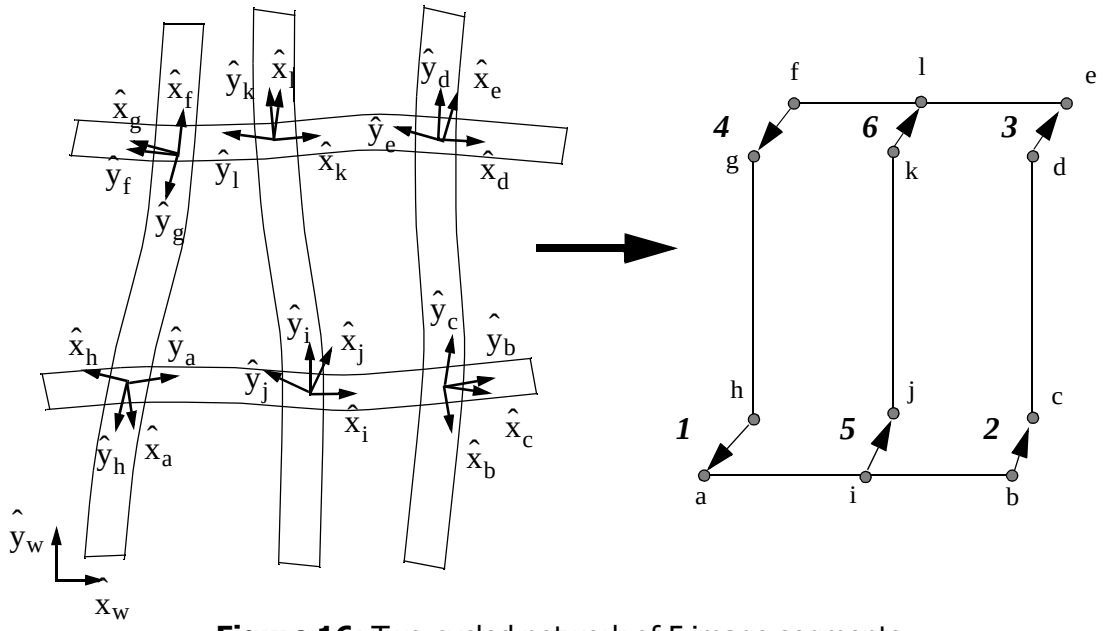


Figure 16: Two-cycled network of 5 image segments

Let the overlaps formed by the pairs *(i,j)* and *(k,l)* contain lockdown points with associated frames *L5* and *L6* respectively. Hence the equation set consists of:

$$T_a^a = T_i^a \cdot t_{L_5}^i \cdot t_j^{L_5} \cdot T_k^j \cdot t_{L_6}^k \cdot t_l^{L_6} \cdot T_f^l \cdot t_{L_4}^f \cdot t_g^{L_4} \cdot T_h^g \cdot t_{L_1}^h \cdot t_e^{L_1} = I_{3\times 3}$$

as well as the equation

$$T_a^a = T_b^a \cdot t_{L_2}^b \cdot t_c^{L_2} \cdot T_d^c \cdot t_{L_3}^d \cdot t_e^{L_3} \cdot T_f^e \cdot t_{L_4}^f \cdot t_g^{L_4} \cdot T_h^g \cdot t_{L_1}^h \cdot t_e^{L_1} = I_{3\times 3}$$

as before, yielding a total of 6 constraint equations in position and orientation.

Differentiating this equation system as before yields an underconstrained system of equations in differential relative pose that can be solved as before using the right pseudo-inverse.

It may be argued that the lockdown graph contains a total of 3 cycles with sequences given by (*a-i-j-k-l-f-g-h-a*), (*a-b-c-d-e-f-g-h-a*), and (*i-b-c-d-l-k-j-i*); and that equations are to be written for all three. In fact, the equations corresponding to the kinematic chains described by the three cycles do not form an independent set. Mathematically, the H matrix formed by using all three equations would be rank-deficient in its rows, and hence the right pseudoinverse would not exist. Resultingly, any two of the three cycles may be chosen in forming the H matrix for the system to be solvable. In other words, the set of cycles that may be used must form an independent and complete cycle cover - a set that, in graph theory, is termed a **fundamental cycle basis**.

9. Mosaicking an arbitrary network of images

9.1 Lockdown Graph Construction

Given the image location of lockdown points in each segment, construction of the lockdown graph corresponding to the network is simply as described below:

```
Start with empty graph
For each segment s
  Set lastSeenImageID = lastSeenLockdownID = -1
  For each image with index i
    If image does not contain a lockdown point, proceed to next image in segment
    If there does exist a vertex with index of the current lockdown point
      Create a new vertex with index of the current lockdown point
    Add a subnode with current image number and segment number to the list of
subnodes of the current vertex
    If (lastSeenImageID != -1)
      Create an undirected edge to the vertex with index = lastSeenLockdownID
and with subnode in the current segment.
    Set lastSeenImageID = i, and lastSeenLockdownID = currentLockdownID
```

9.2 Extraction of Fundamental Cycles

There is considerable amount of literature on finding a fundamental cycle basis of a given graph, given its several applications, including solving electrical networks [1], processing of survey data [2] and others. Deo et al [4] have described several polynomial-time heuristic algorithms for generating a set of fundamental cycles in a graph and have analyzed their performance on the basis of mean fundamental-cycle-set length and execution time for a number of graphs. In our application we simply use a BFS routine, modified along the suggestions in [4], to find a cycle basis for graphs, which in our application are permitted to have multiple self-loops and multiple edges between vertices.

If we weight the edges of a lockdown graph by the number of images constituting the part of the segment between the two lockdown points (or more appropriately, the subnodes on which the two lockdown points lie), it can be seen that the amount of computation involved in our mesh-analysis algorithm is proportional to the length of the cycle set. Unfortunately, Deo et al [4] proved the problem of finding a spanning tree whose fundamental set of cycles has total shortest length to be NP-complete. Horton [11] described a polynomial time algorithm for finding the shortest cycle basis of a graph with *m* number of edges and *n* vertices in $O(m^3n)$ operations, with a formidable worst case of $O(n^7)$. The analysis in [11] was restricted to 2-connected graphs without loops or multiple edges. Since then, Thomassen [24] went on to prove the problem of finding a cycle cover of smallest total length for an arbitrary graph to be NP-hard. The solution to least norm perturbation is, however, not dependent of choice of cycle basis, since the linearized equations corresponding to any cycle basis form an independent set in the same space of constraint equations. The extraction of the smallest basis is therefore not crucial to the algorithm, and is not pursued.

9.3 Computation of the Jacobian for Pose Residuals

Consider a fundamental cycle L represented by a sequence of lockdown point indices, say, $L_1L_2\ldots L_iL_j\ldots L_nL_1$. Consider the portion of the chain between lockdown points L_i and L_j. Let

both lockdown points have a footprint on images in a common segment s, in images k_i and k_j respectively. The expression relating the pose of L_j with respect to L_i will then be of the form

$$(t_{L_i}^{s,k_i})^{-1} T_{k_i+1}^{k_i} T_{k_i+2}^{k_i+1} \dots T_{k_j}^{k_j-1} (t_{L_j}^{s,k_j})$$

or

$$(t_{L_i}^{s,k_i})^{-1} T_{k_i-1}^{k_i} T_{k_i-2}^{k_i-1} \dots T_{k_j}^{k_j+1} (t_{L_j}^{s,k_j})$$

depending on whether $k_i < k_j$ or $k_i > k_j$ respectively. (Recall our convention that the pose of every image is expressed relative to its immediate predecessor in the same segment, having an image index one less than that of the current image, and that the pose of the first image in each segment is expressed relative to the world frame.) Also note that the superscript in the constant t terms denotes both the segment index *s* and the image number. The segment numbers have been dropped in the super- and sub-scripts of the T terms for clarity.

We denote the former case of $k_i < k_j$ as a path of *forward traversal*, and the latter case as that of *backward traversal* of the kinematic chain. The derivation of the Jacobian in both cases is elucidated below.

Case A: Forward traversal

The kinematic expression containing T_{i+1}^{i} is of the form

$$T_i^{L_1} T_{i+1}^{i} T_{L_N}^{i+1} = \begin{bmatrix} R_Q & p_Q \\ \underline{0} & 1 \end{bmatrix} \begin{bmatrix} R_{i+1}^{i} & p_{i+1}^{i} \\ \underline{0} & 1 \end{bmatrix} \begin{bmatrix} R_P & p_P \\ \underline{0} & 1 \end{bmatrix}$$

or the pose residual

$$\begin{bmatrix} R_Z & p_Z \\ \underline{0} & 1 \end{bmatrix} = \begin{bmatrix} R_Q R_{i+1}^{i} R_P & R_Q R_{i+1}^{i} p_P + R_Q p_{i+1}^{i} + p_Q \\ \underline{0} & 1 \end{bmatrix}$$

from which we get two constraints in position as:

$$\frac{\partial p_Z}{\partial p_{i+1}^{i}} = R_Q \qquad \text{and} \qquad \frac{\partial p_Z}{\partial \theta_{i+1}^{i}} = R_Q \dot{R}_{i+1}^{i} p_P$$

where

$$\dot{R}_{i+1}^{i} = \begin{bmatrix} -sin\,\theta_{i+1}^{i} & -cos\,\theta_{i+1}^{i} \\ cos\,\theta_{i+1}^{i} & -sin\,\theta_{i+1}^{i} \end{bmatrix}$$

and the third constraint in orientation as

$$\frac{\partial \theta_Z}{\partial \theta_{i+1}^{i}} = 1$$

Case B: Backward traversal

The kinematic expression containing T_i^{i+1} is of the form

$$T_{i+1}^{L_1} T_i^{i+1} T_{L_N}^{i} = \begin{bmatrix} R_Q & p_Q \\ \underline{0} & 1 \end{bmatrix} \begin{bmatrix} R_i^{i+1} & p_i^{i+1} \\ \underline{0} & 1 \end{bmatrix} \begin{bmatrix} R_P & p_P \\ \underline{0} & 1 \end{bmatrix}$$

i.e. the pose residual

$$\begin{bmatrix} R_Z & p_Z \\ \underline{0} & 1 \end{bmatrix} = \begin{bmatrix} R_Q R_i^{i+1} R_P & R_Q R_i^{i+1} p_P + R_Q p_i^{i+1} + p_Q \\ \underline{0} & 1 \end{bmatrix}$$

which can also be written as

$$\begin{bmatrix} R_Z & p_Z \\ \underline{0} & 1 \end{bmatrix} = \begin{bmatrix} R_Q R_i^{i+1} R_P & R_Q R_i^{i+1} p_P - R_Q R_i^{i+1} p_{i+1}^{i} + p_Q \\ \underline{0} & 1 \end{bmatrix}$$

From this, we get two constraints in position as:

$$\frac{\partial p_Z}{\partial p_{i+1}^{i}} = -R_Q R_i^{i+1} \qquad \text{and} \qquad \frac{\partial p_Z}{\partial \theta_{i+1}^{i}} = R_Q \tilde{R}_i^{i+1} p_P - R_Q \tilde{R}_i^{i+1} p_{i+1}^{i}$$

where

$$\tilde{R}_i^{i+1} = \begin{bmatrix} -sin\theta_{i+1}^{i} & cos\theta_{i+1}^{i} \\ -cos\theta_{i+1}^{i} & -sin\theta_{i+1}^{i} \end{bmatrix}$$

and the third constraint in orientation as

$$\frac{\partial \theta_Z}{\partial \theta_{i+1}^{i}} = -1$$

Concatenating the derivative terms of all the constraint equations yields the underconstrained system of the form described earlier, and the differential pose increment required is solved by the right-pseudo inverse.

Memory Usage and Computational complexity

The approximate size of the H matrix is $3n_l \times 3n_i$ where n_l is the number of loops and n_i is the number of images. Cost of computing HH^T is $O(n_l^2 n_i)$ and cost of inversion is $O(n_l^3)$. In typical networks, $n_l \ll n_i$ and the computational cost of the algorithm for medium to large networks is bounded by a comfortable $O(n_i)$.

10. A Constrained Optimization Approach

The previous section described an efficient technique for map-building in large cyclic environments for observations with small spatial field of view. Each element of the fundamental cycle basis of the topological graph describing the network corresponded to a constraint equation called a loop equation. On the assumption of “local smoothness” with good initial position estimates from odometry, a solution to the required perturbation in relative pose estimates could be obtained at low computational cost. This section presents a formulation that systematically relaxes the above assumption, while preserving the low time complexity of the previous algorithm. The proposed solution employs a total residual minimization approach between observations, while incorporating knowledge of prior state uncertainty and enforcing topological constraints to maintain map consistency.

It is clear that the use of relative poses to describe positions of observations in the map-building problem has several advantages. In mobile robotics applications, it provides a convenient framework to embed initial estimates of relative motion between temporally adjacent observations from a primary position estimation system such as dead-reckoning. Use of relative poses inherently decouples the covariance in non-systematic error between locations of non-temporally adjacent observations, so that the initial state covariance matrix P_0 is of a block-diagonal form. Iterative least-norm perturbation of relative poses to correct map inconsistencies can be done cheaply while fully exploiting topology, while the conversion of observation locations to an absolute reference frame is postponed till after the map-building process is completed.

In a set of observations whose spatial relationships can be described by a tree-like network, the problem of recovering both robot positions and feature locations becomes trivial. This is because the state equations are inherently decoupled, irrespective of the choice of absolute or relative poses for representation, and the state vector can be solved for in time linear in the number of observations. On the introduction of non-temporally adjacent observations, cycles are formed in the topology graph, and the choice of representation becomes crucial in determining time complexity. As seen in Section 7.1 on page 17, the choice of absolute poses requires the inversion of a Hessian matrix having a near block tridiagonal structure with off-diagonal elements corresponding to the points of loop-closure. While this can be achieved in sub-cubic time complexity, inclusion of the dense prior state covariance matrix from a model of odometry makes the problem prohibitively expensive to solve in large scale problems.

At first glance, the choice of relative poses too offers little in the form of favorable matrix structure. On concatenating feature residual observations for tree-like networks, the associated Hessian matrix is block diagonal and invertible in time linear in the number of observations. However, loop-closing observations fill the matrix completely. This is because the absolute pose of one of the images participating in the loop-closing overlap has to be expressed in terms of all the relative poses that precede it in the sequence around a cycle in the topology graph. This causes one row of the Jacobian in the linearized observer equation to fill up, and subsequently the Hessian to be inverted is completely filled.

This trend of filling of matrix sub-blocks suggests the possibility of several avenues for computing estimates of relative pose constrained by the topology of the observations. One possibility, as analyzed earlier, consists of optionally computing the relative poses of temporally

adjacent images while ignoring the loop-closing observations, and then enforcing the topological constraints as a least-norm right-pseudo inverse solution to the basis loop equations. The algorithm was time-efficient but suffered from reliance on accurate values of relative poses being specified at non-temporally adjacent overlaps. Other possibilities include working with a projection of the state space within a smaller state space in which the constraints are automatically enforced, performing SVD on the entire matrix to compute a pseudo-inverse, and enforcing the conditions as weighted artificial observations along with the actual sensor observations. However, all these approaches destroy the inherent structure of the Hessian matrix and make the solution prohibitively expensive.

The solution proposed in this section is a Lagrange multiplier formulation of the map-building problem, where consistency-enforcing loop equations are posed as hard constraints in the system. The state vector consists of elements expressing the change in relative poses between overlapping temporally adjacent images *as well as* those expressing the change in relative poses between overlapping non-temporally images at loop-closure points, and the Lagrange multipliers associated with the constraint equations.

10.1 Priors for Non-temporally Adjacent Overlaps

Equations describing feature residuals at each overlap are expressed in terms of the relative pose connecting the two images participating in the overlap. For instance, in the overlap between images *i* and *j*, the residual of the *m*-th feature as observed in both images is given by

$$z_{ij(m)} = p^{i}_{(i,\,m)} - R^{i}_{j} p^{j}_{(j,\,m)} - t^{i}_{j}$$

where $z_{ij(m)} \in \Re^2$ is the feature residual, $p^{i}_{(i,\,m)} \in \Re^2$ is the position of the *m*-th feature in the i-th image (subscript) represented in the coordinate frame of image *i* (superscript), R^{i}_{j} is the 2-by-2 rotation matrix relating the coordinate frame of image *j* with respect to *i*, and $t^{i}_{j} \in \Re^2$ is the corresponding displacement between the two frames.

On linearizing, this can be expressed as

$$\Delta z_{ij(m)} = -J^{i}_{(j,\,m)} \Delta\rho^{i}_{j}$$

where $J^{i}_{(j,\,m)} \in \Re^{2\times 3}$ is the Jacobian relating the change in position of feature *m* in image *j* with change in relative pose of image *j* with respect to image *i*, and $\Delta\rho^{i}_{j} \in \Re^3$ represents the change in relative pose

Collecting the terms for feature residuals in non-temporally adjacent overlapping images

$$\Delta z_{ij(m)} = -J^{i}_{(j,\,m)} \Delta\rho^{i}_{j}$$

where $j \neq i + 1$, gives a subsystem of the form

$$\Delta \underline{z_{ij}} = M_{ij} \Delta \underline{x_{ij}}$$

which can be solved iteratively by the least-norm left-pseudoinverse solution as

$$\Delta \underline{x_{ij}} = (M^{T}_{ij} M_{ij})^{-1} M^{T}_{ij} \Delta z_{ij}$$

Here the matrix to be inverted for each pair of overlapping images is a 3-by-3 matrix, and hence its computation is inexpensive. Each such solution provides an initial estimate for the relative

pose between non-temporally overlapping images from feature matching. These obtained estimates of relative pose are hence *independent* of the relative pose estimates of all other images in the mosaic. The values of these estimates are used to evaluate the Jacobians used in the overall system matrix described next.

Note that this formulation leaves open the scope of using robust estimators, such as the iteratively reweighted least squares (IRLS) technique commonly encountered in computer vision literature for outlier rejection. There exists a large body of literature on the topic, and further discussion is avoided with the understanding that the techniques can be suitably incorporated in each step involving feature residual minimization using suitable weighting terms.

10.2 Residual Equations for all overlaps

Concatenating residual equations for each feature in each overlap gives a system of equations of the linearized observer form

$$\Delta \underline{z} = H \Delta \underline{x}$$

where the concatenated vector of change in residual is

$$\Delta \underline{z} = \left[\Delta z_{01(0)} \; \Delta z_{01(1)} \; \ldots \; \Delta z_{ij(m)} \; \ldots \; \Delta z_{N-1N(M)} \right]^T$$

and the state vector of change in relative pose is given by

$$\Delta \underline{x} = \left[\Delta \rho_1^0 \; \Delta \rho_2^1 \; \ldots \; \Delta \rho_j^i \; \ldots \; \Delta \rho_N^{N-1} \right]^T$$

Note that the relative poses between temporally adjacent images have initial values from odometry, and those for non-temporally adjacent overlaps are from the feature matching technique described earlier.

10.3 Uncertainty Modeling

The errors in estimate of feature residuals arise primarily from image noise which lead to inaccuracy in specification of corresponding points. This error can be treated as uncorrelated between measurements. We model this error in a loop-closing overlap between images *i* and *j* as a diagonal matrix with entries σ^2, which is our estimate of variance of pixel noise. Then the 3-by-3 error covariance matrix in estimate of relative pose between images *i* and *j* at loop-closure is given by

$$P_{ij} = \sigma^2 (M_{ij}^T M_{ij})^{-1}$$

For temporally adjacent images, the 3-by-3 error covariance matrix corresponding to initial estimate of relative pose from odometry is suitably chosen from an appropriate odometry model [13].

Hence the a priori covariance of the state vector error is given by the block-diagonal matrix P_0 as

$$P_0 = \mathrm{diag}(P_{01}, P_{12}, \ldots, P_{ij}, \ldots P_{N-1N})$$

The matrix is block-diagonal and non-singular, as errors in relative pose are inherently decoupled, and the uncertainty in estimate of non-temporally adjacent relative poses is derived solely from feature matching and not from the poses of other images in the cycle.

10.4 Loop Analysis

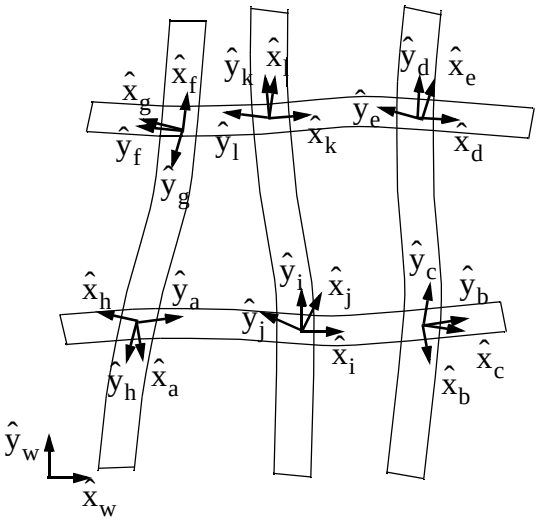


Figure 17a: Two-cycled network of 5 image segments

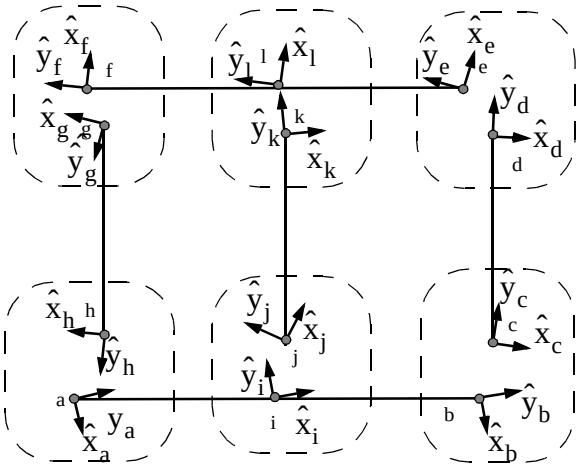


Figure 17b: Topology graph corresponding to the two-cycled network.

Consider the two cycled network shown in figure (17a), composed of 5 overlapping segments. The image pairs (*a*,*h*), (*i*,*j*), (*b*,*c*), (*d*,*e*), (*k*,*l*) and (*f*,*g*) form 6 non-temporally adjacent overlapping image pairs. Let us assume that indices *a* through *h* are in increasing order of magnitude. As per our convention, every image pose is referenced with respect to its immediate predecessor having a smaller image index. Transforms represented by upper-case T refer to unknowns to be determined, but whose initial estimates are available either from odometry, in the case of temporally adjacent images, or from feature matching described earlier, in the case of overlaps between non-temporally adjacent images. A subscript index refers to the frame being referenced, and a superscript index refers to the frame with respect to which the reference is made.

Hence the homogenous transform relating coordinate frame of image *b* to that of image *a* is given by the transform chain

$$T_b^a = T_{a+1}^a T_{a+2}^{a+1} T_{a+3}^{a+2} \ldots T_b^{b-1}$$

Figure (17b) shows an alternate representation of the network, termed a *topology graph* (TG), having the property of having the same number of cycles as the original network of observations. We define a topology graph as a combination of a vertex set υ, and an edge set ε. Each vertex $v \in \upsilon$ is composed of a set of sub-nodes Φ. Each sub-node $n \in \Phi$ belonging to a vertex v represents a unique "footprint" of features associated with v, i.e. the images corresponding to each subnode of a vertex constitute non-temporally adjacent overlaps with common features. Edges between sub-nodes correspond to observed pose relationships.

Like the lockdown graph, the topology graph serves as an intermediate representation of the geometric relationships between image observations for the purpose of extracting constraints implied by them. The major difference between the lockdown graph and the topology graph is the omission of lockdown frames in the latter representation. In the new representation of state, we desire to include relative pose terms between overlapping non-temporally adjacent images in loop-closing overlaps without making a reference to the actual features in common to the overlapping images. In the topology graph, each vertex corresponds to one such overlap, with the

images participating in the overlap appearing as subnodes of the associated vertex. The subnode of one image in the overlap is marked as a *key* subnode. The additional relative poses considered for the overlap are the poses of all other images referenced with respect to the key image/subnode. This marking scheme enables us to avoid redundant inclusion of all relative pose relationships between overlapping images, while at the same time include a minimal but representative number of non-temporally adjacent relative poses for each overlap. These selected relative poses are augmented to the state vector of temporally adjacent relative poses to form the final state vector for the map.

An algorithm for constructing this graph for an arbitrary network of images is as follows:

```
Start with empty graph
For each segment s
  Set lastSeenImageID = lastSeenLockdownID = -1
  For each image with index i
    If image does not contain a lockdown point, proceed to next image in segment
    Set value of currLockdownID to the index of the lockdown point contained by
    the current image
    If vertex with index currLockdownID exists
      Create subnode with index i and add it to the subnode list of the vertex
    else
      Create vertex with index currLockdownID
      Create subnode with index i and add it to the subnode list of the vertex
      Mark subnode with index i to be the key subnode for the vertex, with
      respect to which the relative poses of all other subnodes of the vertex
      are to be referenced.
    If (lastSeenImageID != -1)
      Create an undirected edge to the vertex with index = lastSeenLockdownID
and with subnode in the current segment.
    Set lastSeenImageID = i, and lastSeenLockdownID = currentLockdownID
```

10.5 Constraint Equations

As with the lockdown graph, the homogenous transform expressing a base frame relative to itself, as observed along any cycle in the topology graph, say *(a-b-c-d-e-f-g-h-a)*, should equal the identity. Mathematically,

$$T_a^a = T_b^a \cdot T_c^b \cdot T_d^c \cdot T_e^d \cdot T_f^e \cdot T_g^f \cdot T_h^g \cdot T_a^h = I_{3\times 3}$$

Each equation is a function of the relative poses between images participating in the associated cycle. The set of such equations constitutes the constraint equations that are to be satisfied for consistency. In our example, there are three such equations corresponding to the three possible cycles *(a-b-c-d-e-f-g-h-a)*, *(a-i-j-k-l-f-g-h-a)*, and *(i-b-c-d-l-k-j-i)* in the graph. The linearized forms of any two of these three equations constitute an independent basis in the space of constraints.

10.6 Formulation as a Constrained Estimation

Let each equation in the chosen set of L constraint equations be of the form

$$g_l(X) = b_l$$

where X is the concatenated set of relative poses between temporally adjacent images and non-

temporally adjacent overlapping images.

The maximum likelihood solution to *X* is then a minimizer of the Lagrangian

$$J(X, \lambda) = \frac{1}{2}\sum_{i,j}\sum_{m} z_{ij(m)}^{T} z_{ij(m)} + \frac{1}{2}\sum_{i,j} (\rho_j^i - \hat{\rho}_j^i)^T P_{ij}^{-1} (\rho_j^i - \hat{\rho}_j^i) + \sum_{l=1}^{L} \lambda_l^T [g_l(X) - b_l]$$

which can be represented in matrix form as

$$J(X, \lambda) = \frac{1}{2} z(X)^T z(X) + \frac{1}{2}(X - \hat{X})^T P_o^{-1} (X - \hat{X}) + \lambda^T [g(X) - b]$$

The first term in *J* penalizes the sum of square residuals of feature observations in all overlaps, while the second term penalizes inverse covariance weighted perturbations of relative poses from their estimates. These two terms together constitute a full non-linear optimization that facilitates linearization of the estimation problem at every step using all available observations and the current best estimate of the entire history of the robot trajectory.

The third term containing the vector of Lagrange multipliers λ enforce the loop equations as hard constraints that are to be met by the posterior estimate upto first order accuracy.

The true *X* is a combination of the estimate value $\hat{X}$ and a perturbation ΔX. Hence, the cost function can be rewritten as:

$$J(\Delta X, \lambda) = \frac{1}{2} z(\hat{X} + \Delta X)^T z(\hat{X} + \Delta X) + \frac{1}{2}\Delta X^T P_o^{-1} \Delta X + \lambda^T [g(\hat{X} + \Delta X) - b]$$

Linearizing gives

$$J(\Delta X, \lambda) = \frac{1}{2}[z(\hat{X}) + H\Delta X]^T [z(\hat{X}) + H\Delta X] + \frac{1}{2}\Delta X^T P_o^{-1} \Delta X + \lambda^T [g(\hat{X}) + G\Delta X - b]$$

$$J(\Delta X, \lambda) = \frac{1}{2}[z(\hat{X})^T z(\hat{X}) + 2z(\hat{X})^T H\Delta X + \Delta X^T H^T H \Delta X] + \frac{1}{2}\Delta X^T P_o^{-1} \Delta X + \lambda^T [g(\hat{X}) + G\Delta X - b]$$

Differentiating to extract the Lagrange conditions at point of constrained minima, gives

$$\frac{\partial J}{\partial \Delta X} = H^T z(\hat{X}) + H^T H \Delta X + P_o^{-1} \Delta X + G^T \lambda = 0$$

and

$$\frac{\partial J}{\partial \lambda} = g(\hat{X}) + G\Delta X - b = 0$$

which can be written in matrix form as

$$\begin{bmatrix} H^T H + P_o^{-1} & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} \Delta X \\ \lambda \end{bmatrix} = \begin{bmatrix} -H^T z(\hat{X}) \\ b - g(\hat{X}) \end{bmatrix}$$

Note that this is similar to the general form of the Variable State Dimension Filter (VSDF) [16], and that recursive partitioning can be used to take advantage of the sparseness of the system

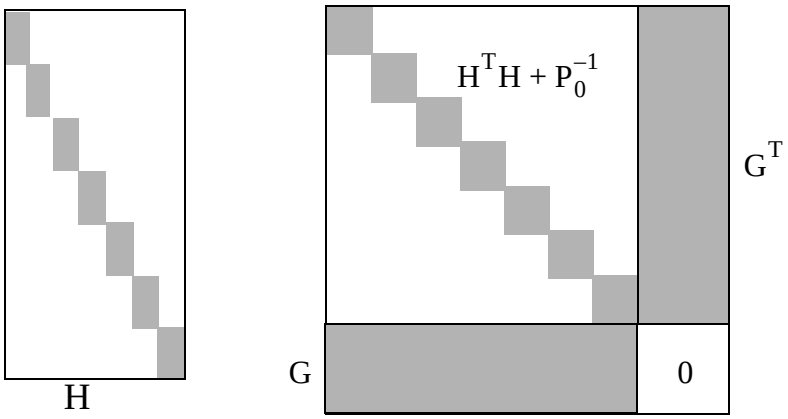


Figure 18: Sparse structure of Jacobian of feature observer H and the matrix of Lagrangian derivatives

matrix to be inverted. By first solving for λ in

$$G(H^T H + P_o^{-1})^{-1} G^T \lambda = [g(\hat{X}) - b] - G(H^T H + P_o^{-1})^{-1} H^T z(\hat{X})$$

the solution for ΔX can be obtained by solving

$$(H^T H + P_o^{-1})\Delta X = -H^T z(\hat{X}) - G^T \lambda$$

Note that the matrix $H^T H + P_o^{-1}$ is block diagonal with a block size of 3, and can hence be inverted in time linear in the number of images.

The above update is performed iteratively until convergence is achieved according to some suitable criterion. The Levenberg-Marquardt procedure can be used to modify this update rule by adding a value ε to the diagonal elements of the linearized system before inversion. This value of ε is increased to smoothly switch towards gradient descent, or decreased to switch towards Gauss-Newton update, depending on whether the value of the cost function *J* increases or decreases respectively.

Computational complexity

The state vector ΔX consists of $n_i - 1$ relative pose terms between n_i temporally adjacent images, and n_k terms corresponding to *k* non-temporally adjacent overlaps. We abbreviate the total number of relative poses in the state vector to be *n*. Let the total number of loops be *l*, and the average number of features considered in each overlap be *p*.

With careful implementation, it can be shown that λ is computable in $O(np + nl + nl^2 + l^3)$ time. Subsequently ΔX can be computed in $O(np + nl)$ time. Hence, the computational complexity of each iteration of the algorithm is bounded by $O(np + nl + nl^2 + l^3)$ where typically $l \ll n$

10.7 Constrained Estimation as a Projection Operation

An interesting observation may be made by setting the Jacobian matrix H to be zero in the linearized Lagrange cost function. This corresponds to performing a batch-step consisting purely of topological constraint satisfaction with knowledge of prior state error covariance P_0. The

resulting system of equations then takes the form

$$\begin{bmatrix} P_0^{-1} & G^T \\ G & 0 \end{bmatrix} \begin{bmatrix} \Delta X \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ b - g(\hat{X}) \end{bmatrix}$$

Solving for ΔX, we get

$$\Delta X = P_0 G^T (G P_0 G^T)^{-1} [b - g(\hat{X})]$$

This is exactly the weighted form of the Moore-Penrose inverse, or least-inverse-covariance weighted norm right pseudo-inverse solution.

so that the new estimate becomes

$$X' = \hat{X} + K_1 [b - g(\hat{X})]$$

where

$$K_1 = P_0 G^T (G P_0 G^T)^{-1}$$

This is the *non-linear constrained estimator* form of the Extended Kalman filter, where K_1 is the Kalman gain matrix of the system. The geometric interpretation of this operation, as pointed out by Newman [17], is that the posterior estimate X' is obtained by projecting the prior estimate X onto a hyper-plane approximation or 'flat' of the true constraint surface $g(X) = b$ computed at $\hat{X}$. Due to linearization assumptions, the constraint equations are met to first-order accuracy by the posterior estimate.

Now consider the original cost function with non-zero H. The expression for λ is obtained by solving

$$G(H^T H + P_o^{-1})^{-1} G^T \lambda = [g(\hat{X}) - b] - G(H^T H + P_o^{-1})^{-1} H^T z(\hat{X})$$

On substituting the expression for λ, the solution for ΔX takes the form

$$\Delta X = -[I - KG][H^T H + P_0^{-1}]^{-1} H^T z(\hat{X}) + K[b - g(\hat{X})]$$

where

$$K = [H^T H + P_0^{-1}]^{-1} G^T [G[H^T H + P_0^{-1}]^{-1} G^T]^{-1}$$

Recall that the estimate of required perturbation in relative pose solely for feature residual minimization is given by the weighted left pseudo-inverse solution

$$\Delta X_r = -[H^T H + P_0^{-1}]^{-1} H^T z(\hat{X})$$

with error covariance of ΔX_r given by $P_r = [H^T H + P_0^{-1}]^{-1}$.

By inspection, it can be seen that the *constrained* estimate of required perturbation in relative pose is of the form

$$\Delta X = [I - KG]\Delta X_r + K[b - g(\hat{X})]$$

which is the constrained estimator form of the Kalman filter, with $K = P_r G^T [G P_r G^T]^{-1}$ being the Kalman gain matrix.

Hence the Lagrange formulation of the map-building problem is a constrained estimator that

1. projects the feature residual vector $z(\hat{X})$ into the column space of H to yield a least-squares unconstrained left pseudo-inverse estimate of perturbation ΔX_r, and
2. projects the unconstrained estimate ΔX_r having covariance P_r onto a linearized hyperplane approximation of the non-linear constraint surface $g(X) = b$.

This sequence of steps thus provides a geometric interpretation of constrained optimization in the map building process.

11. Incorporating Survey Information

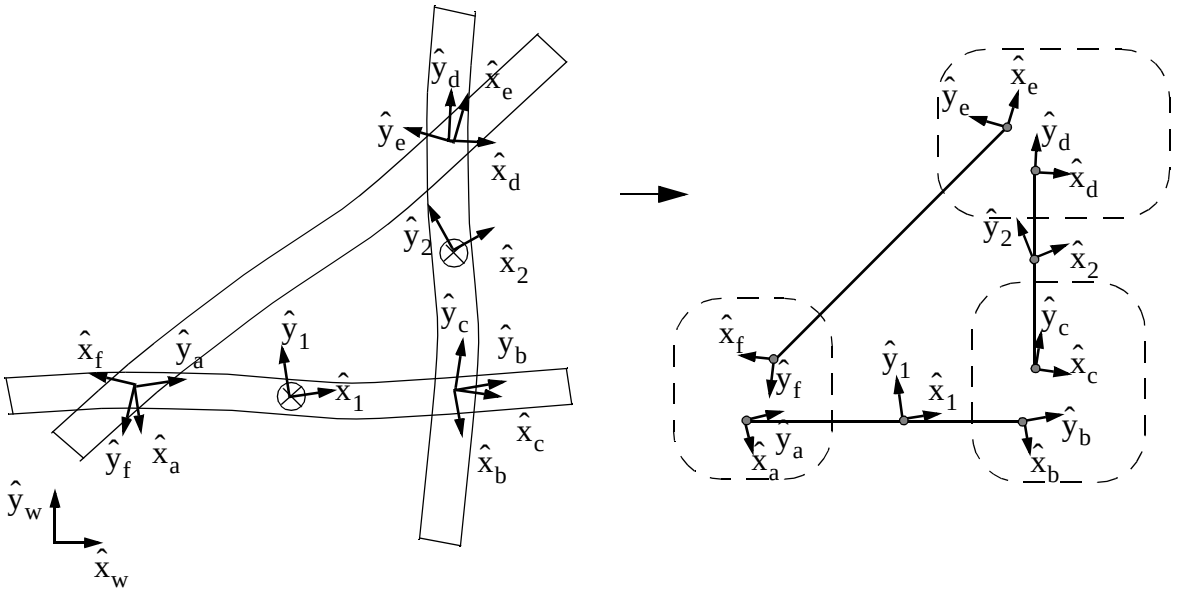


Figure 19: Mosaic with two surveyed fiducials and associated augmented topology graph. Coordinate frames 1 and 2 corresponding to the surveyed measurements are shown.

It is often required to incorporate prior information in the map building process, e.g. data from manual surveying, in order to align internally associated reference frames in the map with some convenient external reference. This section examines the case where surveyed measurements of position and heading are made at fiducials that are observable by the robot. The measurements are assumed to be made with respect to an external world reference frame that is not directly known to the map-building algorithm. Each position in the robot trajectory is represented by a relative pose with respect to the position at the previous time instant. Within this framework of relative poses, map inconsistencies are resolved subject to 3 gauge freedoms - 2 in translation along the world X/Y axes and 1 in rotation.

When one surveyed measurement is supplied, the image containing the surveyed fiducial can be assigned the appropriate value of absolute pose to confirm with the measurement. Absolute poses of other images temporally adjacent to this image can then be set in accordance with the relative poses obtained through loop-analysis after the inconsistencies have been resolved.

Consider the map illustrated in figure (19). It consists of 3 segments overlapping at 3 locations to form one loop. Two surveyed points labeled 1 and 2 are also shown, along with their associated frames. The labeled points are incorporated to form the augmented topology graph shown on the right. The difference between the augmented graph and the topology graph seen earlier is that nodes in the augmented graph can correspond to free (or relative) lockdown points or surveyed points.

Assume that the surveyed measurements of frames 1 and 2, as represented with respect to the (as

yet) unknown world frame, are given by

$$\rho_1^w = \begin{bmatrix} l_{x1} \\ l_{y1} \\ l_{\theta 1} \end{bmatrix} \quad \text{and} \quad \rho_2^w = \begin{bmatrix} l_{x2} \\ l_{y2} \\ l_{\theta 2} \end{bmatrix}$$

respectively.

These constraints can be equivalently represented as the combination of a constraint in absolute position of frame 1, in conjunction with a constraint in relative position of frame 2 with respect to frame 1. i.e.

$$\rho_1^w = \begin{bmatrix} l_{x1} \\ l_{y1} \\ l_{\theta 1} \end{bmatrix} \quad \text{and} \quad \rho_2^1 = \begin{bmatrix} l_{x2}^1 \\ l_{y2}^1 \\ l_{\theta 2}^1 \end{bmatrix}$$

where the "virtual" relative measurements are easily computable through the relation:

$$\begin{bmatrix} l_{x2}^1 \\ l_{y2}^1 \\ l_{\theta 2}^1 \end{bmatrix} = \begin{bmatrix} (l_{x2} - l_{x1}) cos l_{\theta 1} - (l_{y2} - l_{y1}) sin l_{\theta 1} \\ -(l_{x2} - l_{x1}) sin l_{\theta 1} + (l_{y2} - l_{y1}) cos l_{\theta 1} \\ l_{\theta 2} - l_{\theta 1} \end{bmatrix}$$

The relative position ρ_2^1 can be expressed in terms of the relative poses in the map, as a sequence of transformations leading from frame 1 to frame 2.

This sequence is expressed as a **path equation**, as the relative transforms in such a sequence would lie on a path in the lockdown graph starting from node 1 and ending at node 2. This path equation constitutes the extra set of constraints in relative position and heading of frame 2 with respect to frame 1, and is a function of the relative poses of all the images constituting the path.

Consider for example the path 1-b-c-2. The corresponding constraint equations can be written as:

$$T_b^1 T_c^b T_2^c \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} l_{x2}^1 \\ l_{y2}^1 \\ 1 \end{bmatrix}$$

and

$$\theta_b^1 + \theta_c^b + \theta_2^c = l_{\theta 2}^1$$

corresponding to two constraint equations in relative x-y position and one constraint equation in heading, respectively.

Hence by augmenting the matrix of constraint derivatives G, whose rows consist of the Jacobian of the loop equation, with another 3-row consisting of the Jacobian of the path equation above, the system of relative poses can be solved for as before with Lagrange multipliers. This is the solution

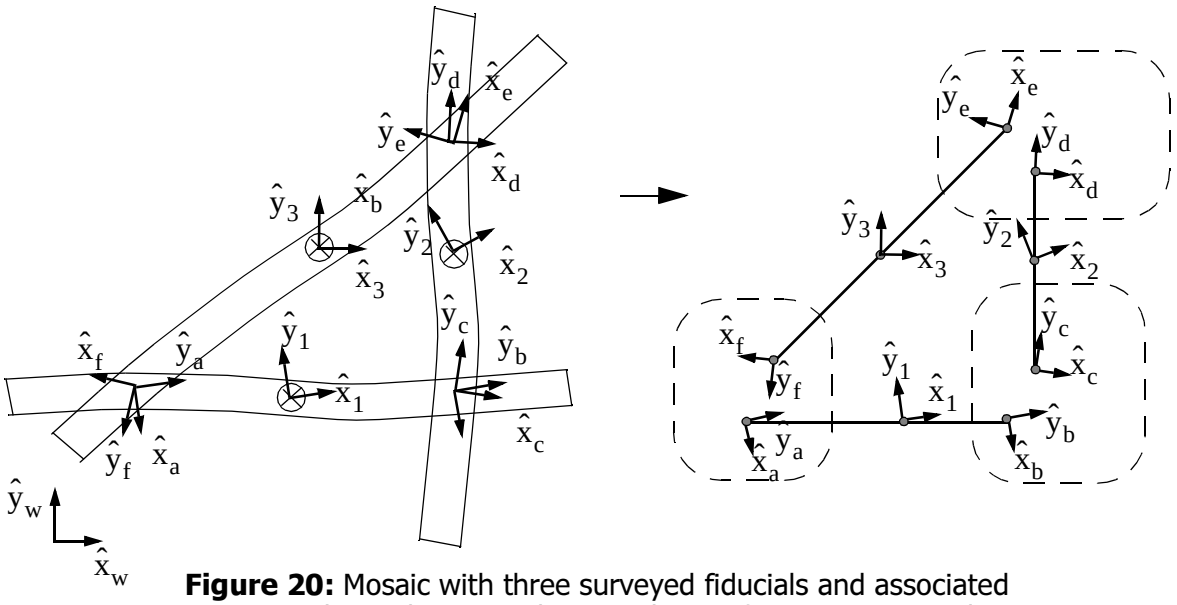


Figure 20: Mosaic with three surveyed fiducials and associated augmented topology graph. Coordinate frames 1, 2 and 3 corresponding to the surveyed measurements are shown.

of the augmented system

$$\begin{bmatrix} H^T H + P_o^{-1} & G_{aug}^T \\ G_{aug} & 0 \end{bmatrix} \begin{bmatrix} \Delta X \\ \lambda \end{bmatrix} = \begin{bmatrix} -H^T z(\hat{X}) \\ b - g(\hat{X}) \end{bmatrix}$$

where

$$G_{aug} = \begin{bmatrix} G_{loop} \\ G_{path} \end{bmatrix}$$

At the end of the iteration, the conversion from relative to absolute poses is initialized by setting frame 1 to that specified by the survey data. It may be noted that the survey pose for both frame 1 and 2 will then be conformed to *exactly.*

Now consider the situation in figure (20) when a third fiducial is surveyed and its measured position and heading information is to be incorporated in the mosaic generation.

This third fiducial, labeled 3, is shown in the map above with its associated frame, as well as in the augmented topology graph as node 3. Let the surveyed pose of this fiducial be given by:

$$\rho_3^w = \begin{bmatrix} l_{x3} \\ l_{y3} \\ l_{\theta 3} \end{bmatrix}$$

This information can be equivalently represented as a constraint in absolute pose of frame 1, along with two constraints in relative pose of frames 2 and 3 with respect to frame 1. This can be

written as

$$\rho_1^w = \begin{bmatrix} l_{x1} \\ l_{y1} \\ l_{\theta 1} \end{bmatrix} \quad \text{and} \quad \rho_2^1 = \begin{bmatrix} l_{x2}^1 \\ l_{y2}^1 \\ l_{\theta 2}^1 \end{bmatrix} \quad \text{and} \quad \rho_3^1 = \begin{bmatrix} l_{x3}^1 \\ l_{y3}^1 \\ l_{\theta 3}^1 \end{bmatrix}$$

The constraint in relative pose of frame 3 with respect to frame 1 can also be represented as a path equation. If the path to be considered is 1-a-f-3, this path equation can be written as

$$T_a^1 T_f^a T_3^f \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} l_{x3}^1 \\ l_{y3}^1 \\ 1 \end{bmatrix}$$

and

$$\theta_a^1 + \theta_f^a + \theta_3^f = l_{\theta 3}^1$$

corresponding to two constraint equations in relative x/y position and one constraint equation in heading, respectively. The G matrix can then be augmented as before, now with two path equations, and the system of relative poses can be solved with the Lagrange multiplier framework.

It may be noted that there exists another constraint in relative pose ρ_3^2 of the frame 3 with respect to frame 2 (or vice versa). However, given the equations expressing ρ_2^1 and ρ_3^1, the relative pose ρ_3^2 is no longer an independent variable. The linearized forms of the path equations constraining ρ_2^1 and ρ_3^1 form an independent basis for the constraints imposed by the surveyed poses. In fact, the equations corresponding to any two of these three variables constitute the largest representative set of constraint equations maintaining full row rank in the G matrix. These equations correspond to the edges of the **spanning tree** containing the nodes in the topology graph corresponding to the surveyed nodes.

Hence in the general case, for observations associated with a topology graph containing n_l independent loops, n_s surveyed fiducials with n_i total observations, there are n_l pose constraints in *loop equations* (global consistency constraints) and $n_s - 1$ pose constraints in *spanning tree equations* (survey constraints). Both types of constraints are treated *simultaneously* and *identically* within the map-building framework.

It follows that the complexity of the mapping algorithm becomes $O(n_c^2 n_i + n_c^3)$ where $n_c = n_l + n_s - 1$ and $n_l \ll n_i$ for sensors with limited FOV.

12. Extension to an Incremental Mapping Algorithm

The ability to decouple the total residual minimization and consistency enforcement steps using a Kalman filter suggests the possibility of an incremental map-building strategy. At each step of incorporation of a new sensor reading there are a finite number of cases to be evaluated, depending on the change enforced in the topology graph.

Let us assume, without loss of generality, that at the current time instant, there are *n* elements in the state vector representing the pose relationships between observations in current (consolidated) map. A (possibly empty) subset n_d of these *n* elements will be a constituent member of one or more loops in the topology graph, and the remainder n_i elements will not be a part of any loop. What this means is that dependencies will exist between n_d relative poses, and the relationships will be described by the loop equations in the current map. The remainder n_i elements will be independent of each other. The state vector can then be partitioned as

$$X = \begin{bmatrix} x_d \\ x_i \end{bmatrix}$$

where subscripts associate the partition with the set of dependent or independent elements, and all vectors are taken by notation to be column vectors.

Let the current estimate of state error covariance be given by $P^{(-)}$. Because of the nature of the partition, this can be written as

$$P^{(-)} = \begin{bmatrix} P_d & 0 \\ 0 & P_i \end{bmatrix}$$

where P_d is a dense $3n_d$-by-$3n_d$ matrix representing the covariance of the n_d elements of relative poses in the dependent set, and P_i is a block diagonal $3n_i$-by-$3n_i$ matrix with 3-by-3 blocks representing the covariance of the n_i elements of relative poses in the independent set. P_i is block diagonal as the random errors in temporally adjacent relative poses that are not part of a loop are uncorrelated.

Let a new observation be made, and the state element corresponding to the relative pose between the current observation and the temporally preceding observation be x_{n+1}

Case A: No feature is observed that is present in a non-temporally adjacent observation. In this case there is no change in the topology graph. The current observation is then simply an extension of a sequence of observations whose associated relative poses are independent. The analysis is done by augmenting the state vector *X* with a term corresponding to the new relative pose and augmenting P_i with another block diagonal element corresponding to the state covariance of the new relative pose, as determined from odometry.

Case B: A feature is observed that is present in a non-temporally adjacent observation with no new loop formed. This happens when an existing map is augmented with information obtained sequentially in a separate mapping run. The relative poses between observations remain independent, but an additional node is created in the topology graph at a location corresponding to the overlap of the two non-temporally adjacent images. The analysis is done by augmenting the state vector *X* with two terms, the first corresponding to the relative pose between the current

observation and the observation temporally preceding it, and the second corresponding to the relative pose between the two non-temporally adjacent images participating in the newly formed overlap. P_i is augmented with a block diagonal element corresponding to the state covariance of the new relative pose, as determined from a model of odometry, and another block diagonal element corresponding to the error covariance in estimate of the relative pose between the overlapping non-temporally adjacent images as determined by the separate feature matching procedure described earlier.

Case C: A feature is observed that is present in a non-temporally adjacent observation and a cycle is formed. This has the effect of adding a new node and a new edge in the topology graph, hence forming a new cycle. In this case, the state vector and covariance matrices are augmented as described in case B, but a separate step of consistency enforcement is carried out. The equation describing the new constraint corresponds to the newly generated cycle in the graph. The new element of the fundamental cycle basis in the modified topology graph can be chosen to be any cycle that contains the newly added edge. As will be shown, greater computational savings are obtained by choosing the smallest such cycle. This smallest cycle is trivially obtained by temporally removing the newly added edge, finding the shortest path between the nodes of the edge in the reduced graph, and declaring the concatenation of the edge and the found path to constitute the shortest cycle.

The state vector can then be reordered and partitioned in the following manner

$$X = \begin{bmatrix} x_m \\ x_l \\ x_a \\ x_b \end{bmatrix}$$

where

x_m consists of the n_m relative pose elements that are part of a previously assimilated cycle in the topology graph, i.e. part of a cycle other than the one newly created,

x_l consists of the n_l relative pose elements that are part of both, a previously assimilated cycle in the topology graph, as well as the newly created cycle,

x_a consists of the n_a relative pose elements that are only part of the newly created cycle and none other, and

x_b consists of the n_b relative pose elements that are not part of any cycle.

It can be observed that before the generation of the new cycle, x_m and x_l together constituted the elements of the dependent set x_d, and x_a and x_b together constitute the elements of the independent set x_i. The corresponding structure of the state covariance matrix can then be expressed as

where P_{aa} and P_{bb} are both block diagonal elements, with each block being a 3-by-3 matrix.

The constrained posterior estimate of the state vector can then be expressed as a projection onto a linearized subspace of the constraint equation.Let the constraint corresponding to the loop equation be $g(X) = b$. Since *g* is not a function of state elements in x_m and x_b, its Jacobian with

$$P^{(-)} = \begin{bmatrix} P_{mm} & P_{ml} & 0 & 0 \\ P_{lm} & P_{ll} & 0 & 0 \\ 0 & 0 & P_{aa} & 0 \\ 0 & 0 & 0 & P_{bb} \end{bmatrix}$$

respect to the state is then given by the form

$$G = \begin{bmatrix} 0 & G_l & G_a & 0 \end{bmatrix}$$

The Kalman "gain" matrix is then given by

$$K = P^{(-)}G^T[GP^{(-)}G^T]^{-1}$$

The term to be inverted can be written as

$$S = GP^{(-)}G^T = \begin{bmatrix} 0 & G_l & G_a & 0 \end{bmatrix} \begin{bmatrix} P_{mm} & P_{ml} & 0 & 0 \\ P_{lm} & P_{ll} & 0 & 0 \\ 0 & 0 & P_{aa} & 0 \\ 0 & 0 & 0 & P_{bb} \end{bmatrix} \begin{bmatrix} 0 \\ G_l^T \\ G_a^T \\ 0 \end{bmatrix}$$

or

$$S = \begin{bmatrix} G_l P_{lm} & G_l P_{ll} & G_a P_{aa} & 0 \end{bmatrix} \begin{bmatrix} 0 \\ G_l^T \\ G_a^T \\ 0 \end{bmatrix} = [G_l P_{ll} G_l^T + G_a P_{aa} G_a^T]$$

It can be observed that the first term $G_l P_{ll} G_l^T$ can be computed in $O(n_l^2)$ time and the second term $G_a P_{aa} G_a^T$ can be computed in $O(n_a)$ time. Since *S* is a 3-by-3 matrix, inversion can be done trivially. The *K* matrix can then be computed as

$$K = \begin{bmatrix} P_{mm} & P_{ml} & 0 & 0 \\ P_{lm} & P_{ll} & 0 & 0 \\ 0 & 0 & P_{aa} & 0 \\ 0 & 0 & 0 & P_{bb} \end{bmatrix} \begin{bmatrix} 0 \\ G_l^T \\ G_a^T \\ 0 \end{bmatrix} S^{-1} = \begin{bmatrix} P_{ml} G_l^T S^{-1} \\ P_{ll} G_l^T S^{-1} \\ P_{aa} G_a^T S^{-1} \\ 0 \end{bmatrix}$$

in $O(n_m n_l + n_l^2 + n_a)$ time.

The updated state estimate is then computed as

$$X^{(+)} = X^{(-)} + K[b - g(X^{(-)})]$$

in a total of $O(n_m n_l + n_l^2 + n_a)$ time.

The updated state covariance is computed as

$$P^{(+)} = P^{(-)} - KGP^{(-)}$$

and is computable in $O((n_m + n_l + n_a)^2)$ time. Although expensive, this computation need only be performed at points of loop closure in the robot's trajectory.

13. Results



Figure 21: (left) Mosaic constructed using only initial pose estimates from dead-reckoning. Note the discrepancy in alignment of the circled feature in the lower left corner at the crossover point. **(right)** Mosaic after consistency enforcement.

Figure (21) shows results with a toy problem of 122 images, using a simulated vehicle moving at 0.5m/s. The actual images used in this experiment are extracted from a larger aerial photograph. Initial estimates of vehicle pose are obtained from a simulated differential heading based dead-reckoning system. Systematic error is introduced by corrupting the velocity measurements of the odometers on the left and right wheels with a scale error of magnitude equal to 5% of the true value measured. The picture on the left illustrates the indication of a rightward drift by dead-reckoning measurements, leading to distortion and inconsistency in the generated map on traversing a loop. The picture on the right shows the map after consistency was enforced with our algorithm.

Figure (22) shows the outlines of the observed vehicle trajectory and the estimated trajectory after map correction through consistency enforcement, as compared to the ground truth trajectory. It can be seen that on the scale of the figure, the corrected and true trajectories are nearly indistinguishable for most of the path. On plotting the image position and orientation residuals in the world reference frame (Figures (23) and (24)), it can be seen that the residual in image heading is bounded to less than 1.5 degrees and that no image deviates from its true position by more than 6 pixels.

Figures (25)-(27) compare the solution obtained with the proposed technique with the solution that enforces only global consistency by loop closure and does not simultaneously enforce local smoothness through feature residual minimization on the assumption of good initial estimates. Figure (25) shows the trajectories corresponding to the two solutions, clearly indicating the solution from the constrained optimization formulation to be more in conformity with the true trajectory. It can also be seen that overall image position residual is lowered (Figure (26)), and that image orientation residuals are comparable and both acceptably small (Figure (27)).

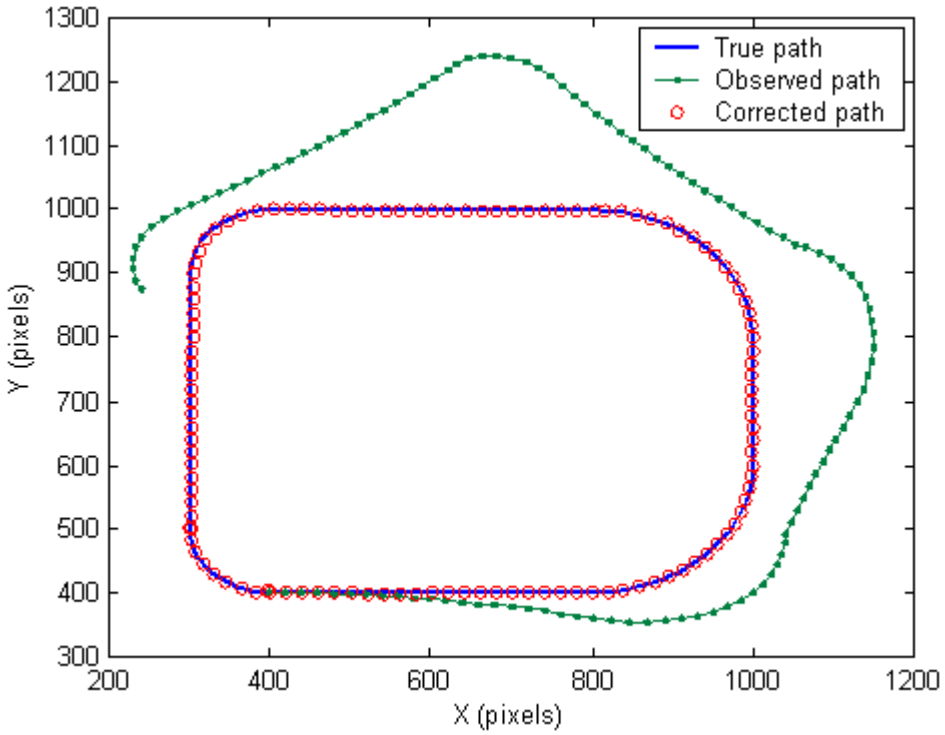


Figure 22: Outline of ground truth, observed and corrected trajectories

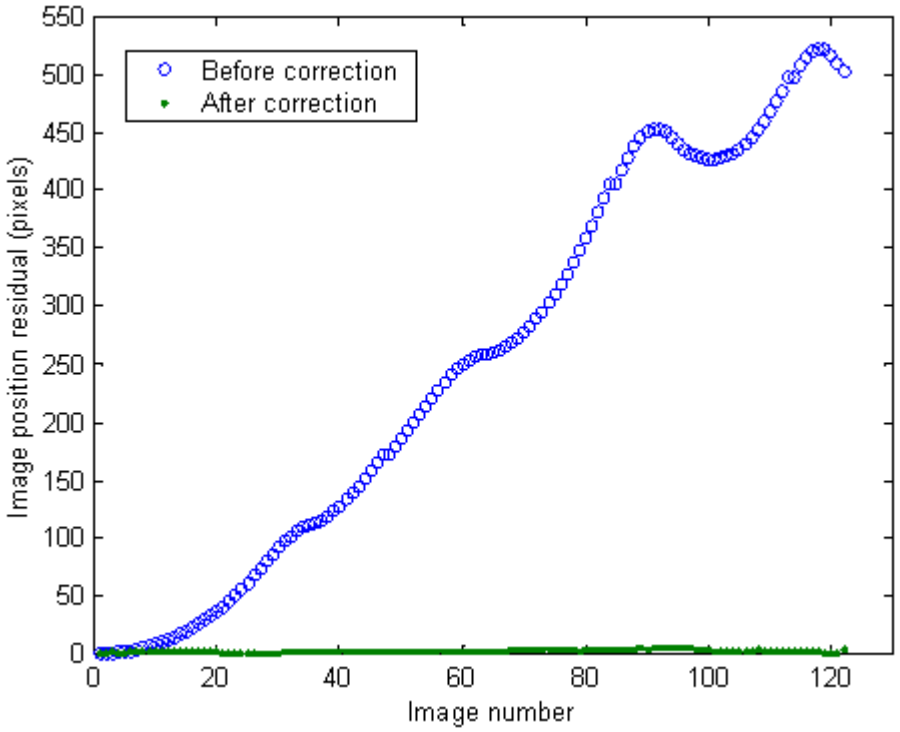


Figure 23: Plot of image position residual with respect to true orientations

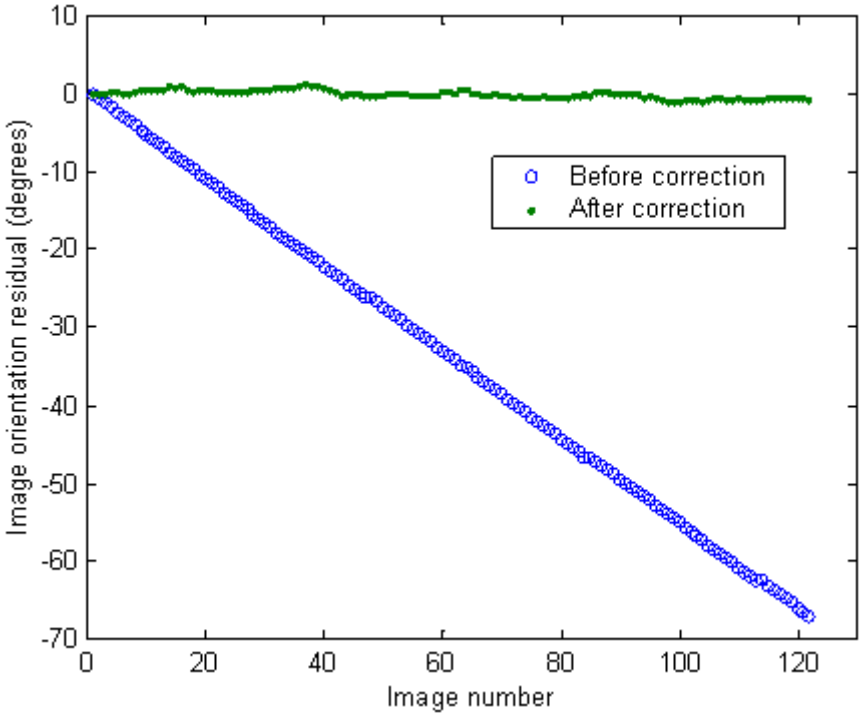


Figure 24: Image orientation residual with respect to true orientation

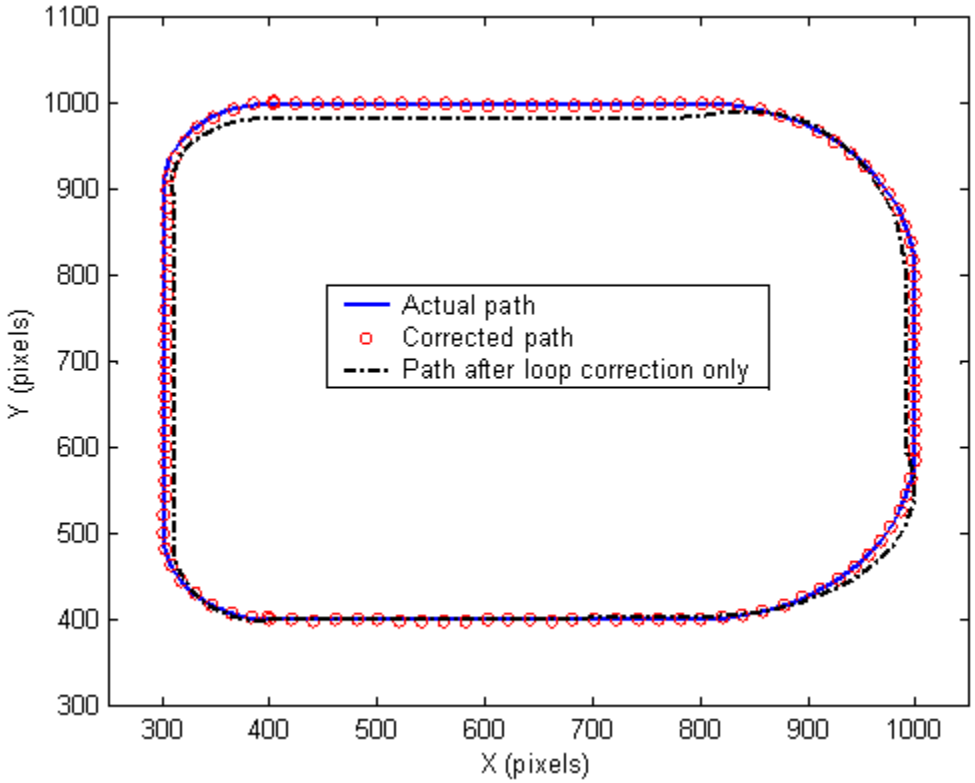


Figure 25: Comparison of corrected trajectories with and without feature residual minimization

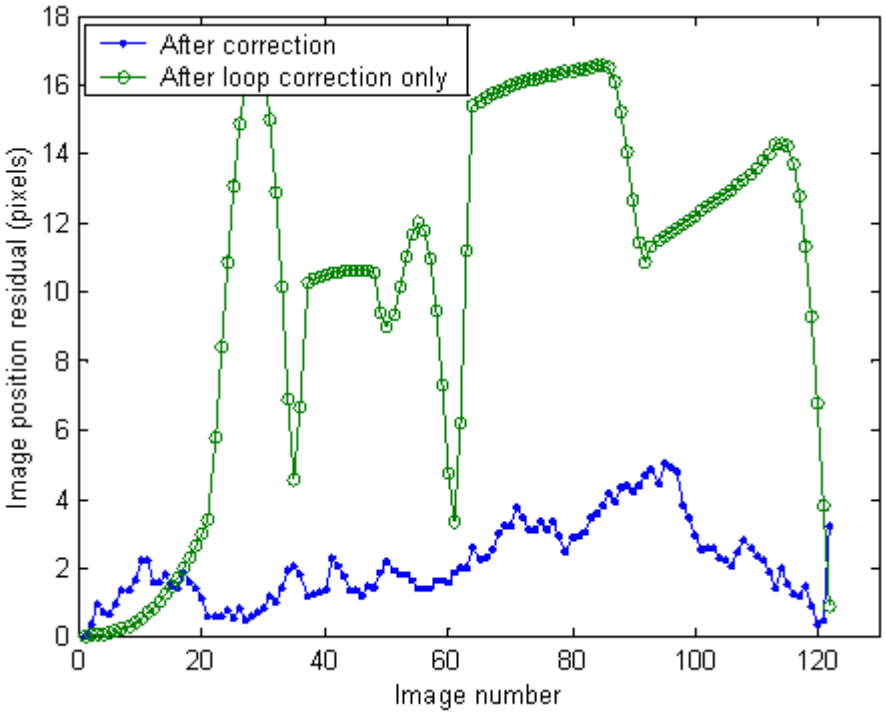


Figure 26: Comparison of image position residual between mosaics corrected with and without feature residual minimization

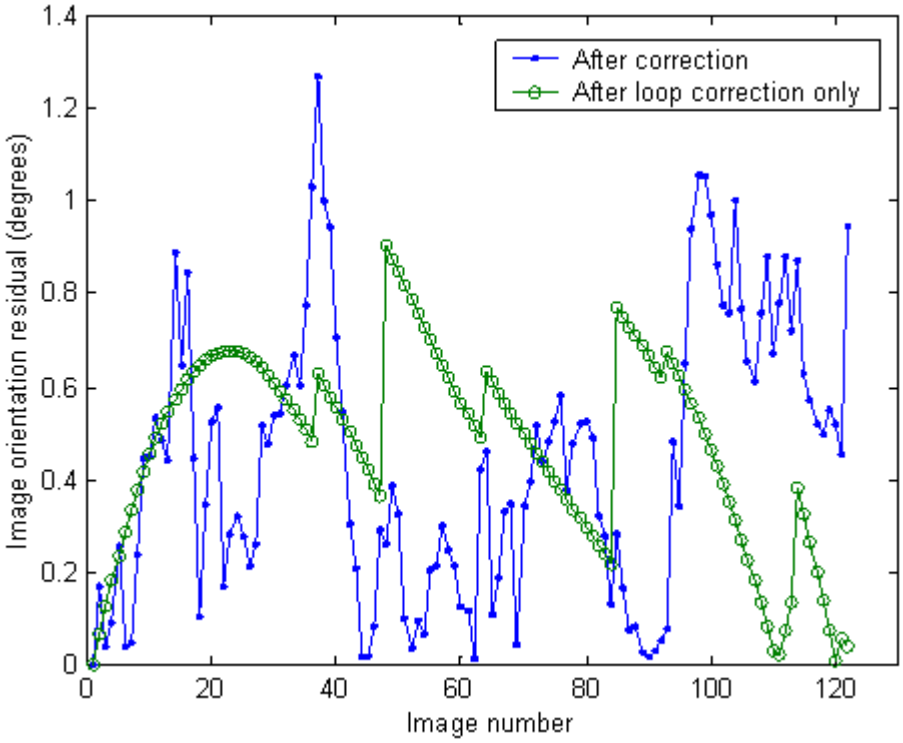


Figure 27: Comparison of image orientation residual between mosaics corrected with and without feature residual minimization

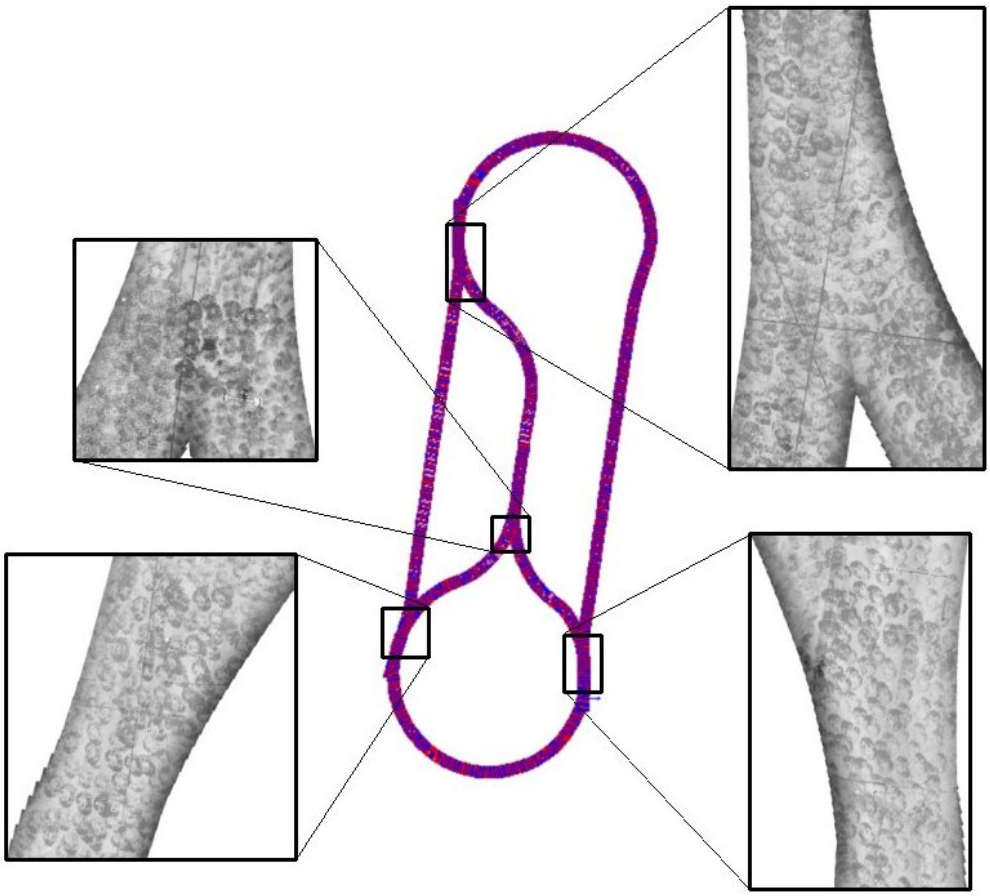


Figure 28: Map of indoor vehicle guidepath. Regions of the final mosaic at the intersections are highlighted.

Figure (28) shows results with a 170m long vehicle guidepath in our indoor laboratory, mapped using our test vehicle with a total of 1836 images. The mosaic built using only initial pose estimates from dead-reckoning had 4 intersections that were initially unresolved. Although our implementation of the algorithms presented in this document were not geared towards optimal computational efficiency, it is worth noting that the algorithm in Section 7.1 based on total residual minimization with absolute pose representation took several hours on a 700Mhz Pentium-III processor, while the algorithm in Section 10.6 based on constrained optimization took less than 10 minutes.

14. Summary and Conclusions

What we have presented is an map-building algorithm that enforces global consistency while minimizing observed feature residuals by exploiting the topology of observations made with sensors having small spatial field of view. The algorithm is $O(np + nl^2 + l^3)$ in complexity, where typically the number of loops *l* is much smaller than the number of images *n*, and hence it scales well to large datasets in the environments common to our application. In our work, we have generated reliable cyclic mosaics ranging from 80m to 800m in total length, both in laboratory and factory environments. The mosaics have been used by our test vehicles at speeds exceeding normal operating conditions for up to 40hr stretches without error requiring human intervention.

The use of a relative pose framework in representing sensor observations has demonstrated several advantages. Position uncertainty between relative poses is inherently decorrelated, and temporary introduction of redundant elements in the state vector to account for correlations in the observations (in our case when loops occur in the mosaic) can make solutions to optimization problems computationally efficient. The disadvantage of using them is in the need to track topology as new observations are introduced and the extra computation required to generate absolute poses from the set of relative poses.

Limited visibility is the defining attribute of an application that makes the presented technique viable and valuable. This attribute is one that is often encountered in indoor applications, where visibility is limited by the geometry of the environment itself, if not by the choice of sensors.

The spatial relationships between observations have embedded topological structure. Correlations between position uncertainties are caused by the dependencies created by loops in the structure. When the mapping problem is posed as one of state estimation, these dependencies appear as constraints in the estimation process. A cycle basis suffices to represent these constraints as loop equations, and can be extracted cheaply using well-known algorithms in graph theory. Additional information from survey data can similarly be represented by spanning tree equations, and be treated identically in the estimation process.

Constrained optimization is a natural formalism for simultaneously enforcing global consistency and local smoothness. The classical Newton's method for optimization generates equations that are solvable by block Gaussian elimination. The use of relative poses and their associated decorrelation make these solvable extremely efficiently.

15. Future Work

The topology graph used in our work serves as a convenient tool for extracting and enforcing constraints in any set of observations, independent of the sensor associated with the observation. Ongoing work includes application of the algorithm in domains with other sensor modalities, including building dense maps with laser-range data, and eliminating the need to survey positions of laser reflectors for the installation of bearing-only guidance systems in factories.

It can be seen that for sensors having larger field of view, the appreciable overlap between observations that are not temporally adjacent but include common features will introduce additional cycles in the topology graph. In the extreme case of all observable environment features being present in every observation, the computational complexity of the algorithm becomes cubic in the number of observations. For many practical applications, however, reliable robot navigation does not necessitate maps with an extremely high local accuracy. Hence, there is often a trade-off that can be made between high computational expense to produce accurate metric maps and cheaply building maps capturing necessary high level topological structure of the environment. In indoor environments, this trade-off can be made by artificially shortening the sensory horizon of the robot in the data collection process. This lowers the computational complexity in map building as the consideration of only features in the near field would reduce the number of cycles in the topology graph. The resultant trade-off in the local metric accuracy of the map would be worth an analytical investigation.

Reliable automated detection of closure points in the robot trajectory and detection of revisited areas is a problem that has only been solved with partial success in the robotics community. Although the domain of our application did not demand real-time mapping capability and allowed for some user intervention, the development of a technique to quickly detect non-temporally adjacent overlapping regions in the mosaic to automatically determine topology is a natural extension to our work. Consistent odometry error modeling and incorporation of online auto-calibration techniques are also being pursued.

16. References

[1] Chua L.O. and Chen L.K., *On optimally sparse cycle and coboundary basis for a linear graph*, IEEE Trans. Circuit Theory, CT-20 (1973), pp. 495-503.

[2] Cribb D.W., Ringeisen R.D. and Shier D.R., *On cycle basis of a graph*, Congr. Numer., 32 (1981), pp. 221-229.

[3] Davis J., "Mosaics of scenes with moving objects", *IEEE Proc. Computer Vision and Pattern Recognition, 1998*, pp. 354 -360

[4] Deo N., Prabhu G.M. and Krishnamoorthy M.S., *Algorithms for generating fundamental cycles in a graph*, ACM Trans. Math. Software, 8 (1982), pp. 26-42.

[5] Durrant-Whyte H.F., "Integration of Disparate Sensor Observations", Proc. IEEE Int. Conf. Robotics and Automation, 1986, p1464.

[6] Fleischer, S.D., Wang, H.H. and Rock, S.M.; Lee, M.J., "Video mosaicking along arbitrary vehicle paths", *Proc. Symposium on Autonomous Underwater Vehicle Technology,* 1996. pp. 293-299

[7] Gracias N., José Santos-Victor, "Underwater video mosaics as visual navigation maps", *Computer Vision and Image Understanding*, Vol. 79(1) pp. 66-91, July 2000

[8] Guivant J. and Nebot E., "Optimization of the Simultaneous Localization and Map Building Algorithm for Real Time Implementation", *IEEE Trans. on Robotics and Automation, June 2001*, 17(3), pp. 242-257

[9] Gutmann J.-S. and Konolige K., "Incremental Mapping of Large Cyclic Environments", *Proc. Computational Intelligence in Robotics and Automation (CIRA)* 1999, pp. 318 -325

[10] Heung-Yeung Shum; Szeliski, R., "Construction and refinement of panoramic mosaics with global and local alignment", *Sixth International Conference on Computer Vision*, 1998 pp. 953 –956

[11] Horton J.D., *A polynomial-time algorithm to find the shortest cycle basis of a graph*, SIAM J.Computing, Vol.16, No,2, April 1987.

[12] Kang E.-Y., Cohen I. and Medioni G., "A Graph-based Global Registration for 2D Mosaics", *15th Intl. Conf. On Pattern Recognition, 2000, Barcelona, Spain*.

[13] Kelly A., "Linearized Error Propagation in Odometry", to appear in the Intl. Journal of Robotics Research, 2002.

[14] Kelly A., "Mobile Robot Localization from Large-Scale Appearance Mosaics", Intl. Journal of Robotics Research, 19(11), 2000

[15] Lu F. and Milios E., "Globally consistent range scan alignment for environment mapping", *Autonomous Robots*, 4: pp. 333-349, 1997

[16] McLauchlan P.F., "The Variable State Dimension Filter applied to Surface-based Structure from Motion", *CVSSP Technical Report VSSP-TR-4/99*

[17] Newman P., "On the Structure and Solution of the Simultaneous Localization and Map Building Problem", *Ph.D. thesis submitted in March 1999, Australian Centre for Field Robotics,*

Univ. of Sydney.

[18] Reddy B.S. and Chatterji B.N., "An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration", *IEEE Trans. Image Processing*, Vol. 5, No. 8, August 1996

[19] Rowe P. and Kelly A., "Map Construction for Mosaic-Based Vehicle Position Estimation", *International Conference on Intelligent Autonomous Systems (IAS)*, July, 2000.

[20] Sawney H.S., Hsu S. and Kumar R., "Robust video mosaicking through topology inference and local to global alignment", In *Proc. European Conference on Computer Vision, ECCV, Freiburg Germany,* vol.2, pp. 103-119, June 1998

[21] Sawhney H.S., Kumar R., Gendel G., Bergen J., Dixon D., and Paragano V., "VideoBrush: experiences with consumer video mosaicking", *Proc. Fourth IEEE Workshop on Applications of Computer Vision (WACV), 1998*. pp. 56 –62

[22] Smith R., Self M. and Cheeseman P., "Estimating uncertain spatial relationships in robotics", In Cox and Wilfong, *Autonomous Robot Vehicles*, Springer-Verlag 1990, pp. 167-193.

[23] Szeliski, R.; Coughlan, J., "Hierarchical spline-based image registration", *IEEE Proc. Computer Vision and Pattern Recognition*, 1994, pp. 194 –201

[24] Thomassen C., "On the complexity of finding a minimum cycle cover of a graph", *SIAM J.Computing*, Vol.26, No.3, pp. 675-677, June 1997

[25] Unnikrishnan R. and Kelly A., "Mosaicking Large Cyclic Environments for Visual Navigation in Autonomous Vehicles", *Proc. IEEE Conf. on Robotics and Automation (ICRA), Washington D.C., May 2002*.

[26] Unnikrishnan R. and Kelly A., "A Constrained Optimization to Globally Consistent Mapping", *Proc. IEEE Conf. on Intelligent Robots and Systems (IROS), Lausanne, Switzerland, October 2002*.