Mosaicing Large Cyclic Environments for Visual Navigation in Autonomous Vehicles

Ranjith Unnikrishnan and Alonzo Kelly

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
email: ranjith@andrew.cmu.edu, alonzo@ri.cmu.edu

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 locally smooth and globally consistent high-resolution mosaic of the vehicle's environment, efficiently done using observations that have low spatial and temporal persistence. 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 paper presents a method that reliably generates consistent maps at low computational cost, while fully exploiting the topology of the observations. Extensions to a real-time implementation are discussed along with results using simulated data and those from real indoor environments.

1 Introduction

Navigating from imagery is 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 [10] is a technique employing real-time imagery to track motion over a previously stored high-resolution image of a known environment. By reducing the localization problem to one of template matching, it provides absolute position fixes which are used to damp the growth of errors that occur in a primary position estimation system such as odometry. Previous papers on the subject of mosaic-based position estimation have discussed the feasibility of such an approach [10], provided a description of the tracking algorithm used to navigate the mosaic [9], and a basic algorithm to construct the mosaic itself [14].

A *guidepath* is a portion of a known vehicle trajectory. A set of possibly intersecting guidepaths is called a *guidepath network*. Each piece of mosaic associated with an edge of the guidepath network of the vehicle, as collected during a single swath of the vehicle is referred to a *map segment*. The images used to create a mosaic are collected into larger one-dimensional segments by passing a camera over the floor and recording both the imagery and the locations of each image, as determined through dead-reckoning. The mosaic is hereafter referred to as the *map*.

1.1 Issues with Map Construction

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 high-speed visual tracking to be possible, the constructed map must satisfy the conditions of *local smoothness*, whereby the error in temporally adjacent images is locally bounded to some acceptably small value, and *global consistency*, whereby the position reported at a particular location becomes neither time nor path dependent, and is uniquely represented in the constructed model.

These requirements are common to any process involving the construction of reliable maps, 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.

1.2 Prior Mosaicing work

The field of image mosaicing is a relatively old one, with no dearth of research in automated mosaicing or its applications. Several methods have been proposed, including the solution of a linear system derived from the collection of pair-wise registration matrices [2], or the frame-to-mosaic scheme [7]. Only more recently have near real-time [16] and globally-consistent [15] solutions emerged. Recently, Kang et al. [8] 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 mosaiced.

1.3 Map Building in Cyclic Environments

Considerable research has been done in the field of real-time video mosaicing of the ocean-floor for navigation, exploration and wreckage visualization. Gracias and Santos-Victor [12] present several algorithms based on a projective geometry framework, and using robust matching techniques for frame-to-frame alignment. Work by Fleischer et al [5] 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 [14] 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 [11] recognized the need for a simultaneous solution in their work on automated mapping with a laser rangefinder. 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 [6] presented what they believe to be the first 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 mosaicing application, because 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.

This paper presents a technique for cost-effectively mapping cyclic network environments from low-visibility observations, and 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.

2 Linear mosaicing

Linear mosaics are the simplest mosaics to construct. A smooth linear mosaic can be constructed simply by sequentially registering images at each overlap as it is formed, so as to minimize the feature position residual in the mosaic as it is traversed from start to finish. This problem is solvable in insignificant time [13][18] because the unknowns are not coupled at all. The linear mosaic, or segment, is hence locally smooth, as the maximum distortion error due to incorrect pose fits is small and bounded.

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.

2.1 Total residual of a linear mosaic

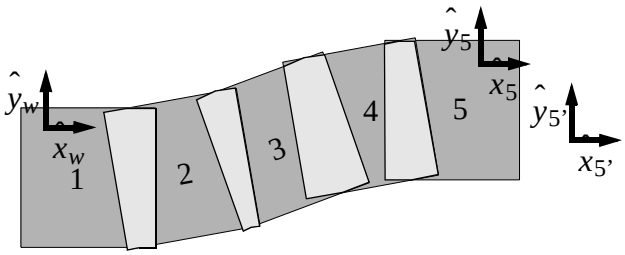

Figure 1: Linear mosaic distortion

A slightly more difficult problem than linear mosaicing is the problem of distorting a linear mosaic so that its endpoint is moved a small amount in position and orientation, so as to confirm to some known reference. Consider the situation outlined in Figure 1 where a linear sequence of images overlap such that 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 *i* in increasing order of image capture, with absolute pose ρ_i^w expressed with respect to a world frame *w*.

Suppose it is necessary to move image frame 5 to pose 5' in a manner which minimizes the *total residual* of the point features at each image overlap. Let at least two point features be identified and their new locations be used to constrain the pose of frame 5' relative to frame 5. 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 change in the absolute pose of image *i+1*. Then the linearized change in residual in each overlap feature can be expressed as a function of the Jacobian and the change in the absolute pose 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:

$$\Delta z_m^5 = -{}_5J_5^m \Delta\rho_5^w = \begin{bmatrix} {}_5J_5^m \end{bmatrix} \begin{bmatrix} \Delta\rho_5^w \end{bmatrix}$$

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:

$$\Delta z = H \Delta x$$

where Δx is a column vector of adjoined absolute pose changes, and Δz is a column vector of adjoined changes in feature residuals. Let the state covariance matrix, as estimated from a suitable model of odometry, be $P \in \Re^{3n_i \times 3n_i}$ for n_i images. Note that the uncertainty in the absolute pose of an observation, whose initial estimate is obtained from dead-reckoning, will be dependent on the uncertainty in estimates of the absolute poses of preceding observations. Hence, the matrix *P* will be dense.

The overdetermined system of equations are solvable using the left pseudoinverse least-squares solution:

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

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 in *H* reduce to zero and can be removed from the matrix altogether. This leaves the *H* matrix to be of size $2m \times 3(n_i - 2)$, where *m* is the total number of feature points.

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 has approximately $4n_f \times 3n_i = 12n_f n_i$ non-zero elements. *P* similarly has $n^2/2$ elements, and these can easily become unmanageably large numbers. The cost of computing $H^T H$ is $O(n_f n_i)$. The cost of inversion of $H^T H + P^{-1}$ is $O(n_i^3)$ and typically dictates the overall cost of this approach.

A commonly-made simplifying assumption is that the true absolute poses are independent gaussian noise-corrupted versions of the initial estimates of absolute pose. This effectively makes the P matrix diagonal, and subsequently the $H^T H + P^{-1}$ matrix to be inverted takes a near block-diagonal form with a half-band width of 5. Efficient inversion techniques for matrices of this type are well-studied, and may be used to solve the equations with sub-cubic time complexity in n_i. However, such an unrealistic error model assumption weights perturbations in absolute pose of each observation equally, leading to inaccurate warping of the mosaic and slower overall convergence.

A solution of this general form does not scale well to a network of images. Overlaps between non-temporally adjacent images introduce off-diagonal block elements in the adjoined Jacobian matrix. These entries are crucial to the generation of a globally consistent mosaic, but destroy the near block-diagonal structure of the $H^T H$ matrix. This in turn makes the problem of mosaicing several thousands of images 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.

2.2 Smooth distortion of a linear mosaic

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 now choose to associate each image *i* by its pose ρ_i^{i-1} expressed *relative* to the preceding image in its segment.

Let two features, *l* and *m*, be identified and their new locations be used to constrain the pose of frame $5'$ relative to frame 5. Let ${}_l J_i^{i-1}$ denote the Jacobian relating the perceived change in absolute position of feature *l* with change in relative pose ρ_i^{i-1}.

This change in residual can be written as:

$$\Delta z_l^1 = \begin{bmatrix} {}_l J_1^2 & \cdots & {}_l J_{i-1}^i & \cdots & {}_l J_4^5 \end{bmatrix} \Delta x$$

where Δx is the vector of concatenated relative poses for images 1 to 5.

The second feature point, *m*, can be interpreted as constraining the orientation of frame 5. The change in orientation residual can be expressed as:

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

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 $\Delta\theta_m^5$ represents the change in relative orientation of the feature *m* in frame $5'$, which is zero since frame $5'$ is fixed.

The adjoined equations can be represented as:

$$\begin{bmatrix} \begin{bmatrix} \Delta x_l^1 \\ \Delta y_l^1 \end{bmatrix} \\ \Delta\theta_m^1 \end{bmatrix} = \begin{bmatrix} {}_l J_1^2 & \cdots & {}_l J_{i-1}^i & \cdots & {}_l J_4^5 \\ \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} & \cdots & \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} & \cdots & \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \end{bmatrix} \begin{bmatrix} \Delta\tilde{\Re}_{1\ldots5} \end{bmatrix}$$

which is of the form:

$$\Delta z = H \Delta x$$

Let the state covariance matrix, as estimated from a suitable model of odometry, be $P \in \Re^{3n_i \times 3n_i}$ for n_i images. Note that the uncertainty in relative pose of an observation, whose initial estimate is obtained from dead-reckoning, will be independent of the uncertainty in estimates of the relative poses of preceding observations. Hence, the matrix P will be block diagonal with block size 3-by-3. The equations are underdetermined, and will therefore be solvable by the general right pseudoinverse, or Moore-Penrose inverse as:

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

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 essential that no more features be considered in image 5 and 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. It is also noteworthy that the residual after the correction of relative poses is zero, as there is an exact solution to the underdetermined system.

Memory Usage and Computational complexity

It may be observed that for a linear mosaic of n_i images, the size of the *H* and *P* matrices is $O(n_i)$ elements. The cost of computing HPH^T is only $O(n_i)$, and being of size 3-by-3 can be inverted in constant time. Resultingly the order of the whole process is $O(n_i)$.

3 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 pose of images participating in an intersection in terms of the relative poses of all images encountered on traversing any path from a base image to the overlap, we can compute minimal changes in image positions for a required affected change in end image pose. For a network, this means solving for perturbations in relative poses, subject to constraints imposed by the topological relations between the observations themselves [4].

Example 1: **One-cycle topology**

Consider the simple single-cycled network shown in Figure 2. The images pairs *(h,a), (b,c), (d,e)* and *(f,g)* form four overlaps, and each image in a pair is registered to its counterpart. This means that the pose of one image relative to the other in each pair is known either through manual specification or through an automated prior stage of checking and appropriately registering candidate pairs of images that are in sufficient proximity of each other to constitute a potential overlap.

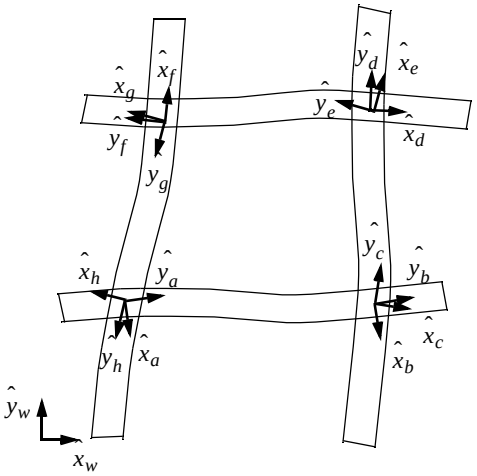

Figure 2: One-cycled network of 4 image segments

Let us refer to a *lockdown point* as a feature that encodes both position and orientation information. The 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. The observed pose of the lockdown point with respect to each of the two image frames are related by the previously determined relative pose between the two image frames. We can thus assign each overlap with a unique virtual lockdown point, or equivalently with an associate coordinate frame termed the *lockdown frame*. Let the image pairs *(h,a), (b,c), (d,e)* and *(f,g)* have lockdown frames *L1*, *L2*, *L3* and *L4* respectively.

Let us assume for now that image indices corresponding to the letters *a*, *b*, *c* and *d* are in increasing order. As per the convention we will adopt, every image pose is referenced with respect to its immediate predecessor having a smaller image index. Homogenous transforms represented by lower-case t refer to constant quantities, such as relationships between images that are determined by a prior registration process, or the relationship between a lockdown point and the image that contains it. Transforms represented by upper-case T refer to unknowns which are to be determined, but whose initial estimates are available from odometry. 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 frame *b* to *a* can be written as:

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

and similarly for *d* to *c*, etc.

Figure 3 shows an alternate representation of the network, termed a **lockdown graph**, which has the property of having the same number of cycles as the original network of observations. We define a lockdown graph as a combination 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 sub-nodes Φ. Each sub-node $n \in \Phi$ belonging to a vertex v represents a unique "footprint" of the lockdown frame associated with v, i.e. there exists a sub-node corresponding to each image of the overlap set that contains the lockdown point.

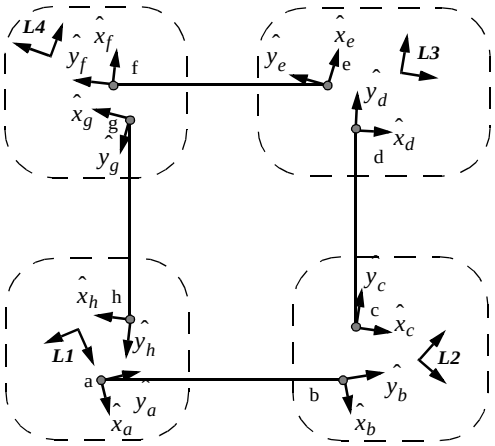

Figure 3: Lockdown graph corresponding to a one-cycle network.

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 observed along the cycle *(a-b-c-d-e-f-g-h-a)*, should equal identity.

Mathematically,

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

or

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

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 equations that comprise a system of the form

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

where Δx is the augmented vector of change in relative poses of all the images in the network, and Δz is the pose residual observed along the kinematic chain of the loop broken at *a*. This system is underconstrained, and its solution is a constrained estimate of least-norm perturbation given by the right pseudo-inverse as

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

The posterior estimate is thus the normal projection of the prior estimate onto a hyperplane approximation of the constraint surface described by the loop equation.

Example 2: **Two-cycle topology**

Consider Figure 4 where an additional segment has been incorporated so as to generate an additional cycle in the topology

graph. This cycle addition generates another equation of the form of that in Example 1.

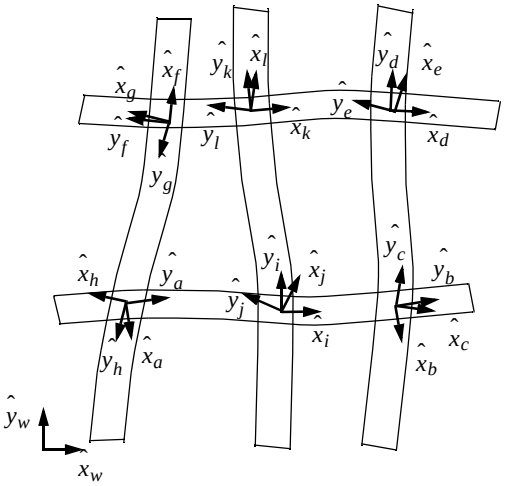

Figure 4: Two-cycled network of 5 image segments

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:

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

from the cycle *(a-i-j-k-l-f-g-h-a)*, 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 4 constraint equations in position and 2 in 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 topology 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. Hence, 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**.

4 Mosaicing Arbitrary Networks (N-cycle topology)

4.1 Construction of lockdown graph

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 sub-node with current image number and segment number to the list of sub-nodes of the current vertex.
    If (lastSeenImageID != -1),
      Create an undirected edge to the vertex with index = lastSeenLockdownID and with sub-node in the current segment.
    Set lastSeenImageID = i, and lastSeenLockdownID = currentLockdownID.

4.2 Extraction of fundamental cycles

There is considerable literature on finding a fundamental cycle basis of a given graph, given its several applications, including solving electrical networks [1], processing of survey data and others. Deo et al [3] 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 work we use a simple breadth-first-search routine, modified along the suggestions in [3], to find a cycle basis for graphs that are allowed to have multiple self-edges and multiple edges between vertices.

If the edges of a lockdown graph are weighted by the number of images constituting the part of the segment between the two lockdown points (or more appropriately, the sub-nodes on which the two lockdown points lie), choosing the smallest-sized cycle basis will reduce computational cost in computing HH^T to a small extent. This problem of finding a cycle cover of smallest total length for an arbitrary graph has been proven to be NP-hard [19]. 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.

4.3 Computation of the Jacobian for pose residuals

Consider a fundamental cycle L represented by an ordered sequence of lockdown point indices, say, $L_1 L_2 \ldots L_i L_j \ldots L_n L_1$. Consider the portion of the chain between lockdown points L_i and L_j, which are adjacent in the sequence. 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

$$\left(t_{L_i}^{s,k_i}\right)^{-1} T_{k_i+1}^{k_i} T_{k_i+2}^{k_{i+1}} \cdots T_{k_j}^{k_j-1} \left(t_{L_j}^{s,k_j}\right)$$

or

$$\left(t_{L_i}^{s,k_i}\right)^{-1} T_{k_i-1}^{k_i} T_{k_i-2}^{k_{i-1}} \cdots T_{k_j}^{k_j+1} \left(t_{L_j}^{s,k_j}\right)$$

depending on whether $k_i < k_j$ or $k_i > k_j$ respectively. 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. We derive the Jacobian of an observation equation with respect to the term ρ_{i+1}^i for both cases.

Case A: Forward traversal

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

$$T_i^{L_1} T_{i+1}^{i} T_{L_N}^{i+1} = \begin{bmatrix} R_Q & r_Q \\ \underline{0} & 1 \end{bmatrix} \begin{bmatrix} R_{i+1}^{i} & r_{i+1}^{i} \\ \underline{0} & 1 \end{bmatrix} \begin{bmatrix} R_P & r_P \\ \underline{0} & 1 \end{bmatrix}$$

or the pose residual

$$\begin{bmatrix} R_Z & r_Z \\ \underline{0} & 1 \end{bmatrix} = \begin{bmatrix} (R_Q R_{i+1}^{i} R_P) & (R_Q R_{i+1}^{i} r_P + R_Q r_{i+1}^{i} + r_Q) \\ \underline{0} & 1 \end{bmatrix}$$

from which we get two constraints in position as:

$$\frac{\partial r_Z}{\partial r_{i+1}^{i}} = R_Q \qquad \text{and} \qquad \frac{\partial r_Z}{\partial \theta_{i+1}^{i}} = R_Q \dot{R}_{i+1}^{i} r_P$$

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

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

$$T_{i+1}^{L_1} T_i^{i+1} T_{L_N}^{i} = \begin{bmatrix} R_Q & r_Q \\ \underline{0} & 1 \end{bmatrix} \begin{bmatrix} R_i^{i+1} & r_i^{i+1} \\ \underline{0} & 1 \end{bmatrix} \begin{bmatrix} R_P & r_P \\ \underline{0} & 1 \end{bmatrix}$$

or the pose residual

$$\begin{bmatrix} R_Z & r_Z \\ \underline{0} & 1 \end{bmatrix} = \begin{bmatrix} (R_Q R_i^{i+1} R_P) & (R_Q R_i^{i+1} r_P + R_Q r_i^{i+1} + r_Q) \\ \underline{0} & 1 \end{bmatrix}$$

which can also be written as

$$\begin{bmatrix} R_Z & r_Z \\ \underline{0} & 1 \end{bmatrix} = \begin{bmatrix} (R_Q R_i^{i+1} R_P) & (R_Q R_i^{i+1} r_P - R_Q R_i^{i+1} r_{i+1}^{i} + r_Q) \\ \underline{0} & 1 \end{bmatrix}$$

From this, we get two constraints in position as:

$$\frac{\partial r_Z}{\partial r_{i+1}^{i}} = -R_Q R_i^{i+1}$$

and

$$\frac{\partial r_Z}{\partial \theta_{i+1}^{i}} = R_Q \tilde{R}_i^{i+1} r_P - R_Q \tilde{R}_i^{i+1} r_{i+1}^{i}$$

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 computed with the right pseudoinverse.

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 fundamental loops and n_i is the number of images. P is of block diagonal form and is described by $9n_i$ elements. Cost of computing HPH^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 $O(n_l^2 n_i)$.

4.4 Incorporating external information

It is possible to incorporate information available through peripheral measurements in the map creation process by expressing the information as a known relationship between observations that contain the related features. If, for instance, the observed image *a* contains a feature s_a and observed image *b* contains s_b, and that the two features are separated by a known (perhaps surveyed) distance, say *d*. We can introduce nodes for the two images, say L_a and L_b, in the lockdown graph, corresponding to their position in the segments containing each of the two images. Any path from L_a to L_b in the lockdown graph can be then associated with a unique sequence of transformations that expresses one image frame which respect to the other.

The linearized form of this expression relating the position of frame *a* with respect to frame *b* constitutes another constraint equation, or another row in the *H* matrix. Subject to the condition that this path does not close to form a cycle, the inclusion of this additional equation incorporates extrinsically available information between non-temporally adjacent images, without introducing rank deficiency in matrix *H*. This can be used to enforce external conformity to known landmark positions in the world frame, or retain known spatial relationships between non-adjacent observations of different features.

5 Results

Figures 5-8 show results from images captured by a vehicle, simulated at 1/10th scale, moving at 0.5m/s with a differential-heading odometer calibrated to 99% accuracy in wheel velocity

Figure 5: Mosaic constructed using only odometry-based initial pose estimates. Note the discrepancy in alignment of the circled feature at the crossover point (lower left corner).

measurements. In this toy problem consisting of 121 images, the figure illustrates the indication of a systematic rightward drift by dead-reckoning measurements. It can be observed that the result from correction based on loop-analysis are very comparable to that based on total-residual minimization. While the total residual is less in the latter, local smoothness is shown to be preserved to a greater extent by our proposed method. The absolute heading error in the final mosaic is also bounded to less than 1°.

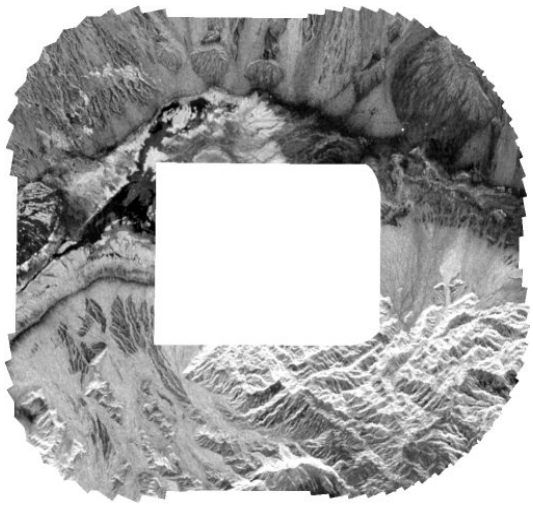

Figure 6: Mosaic after cycle closure with loop-analysis

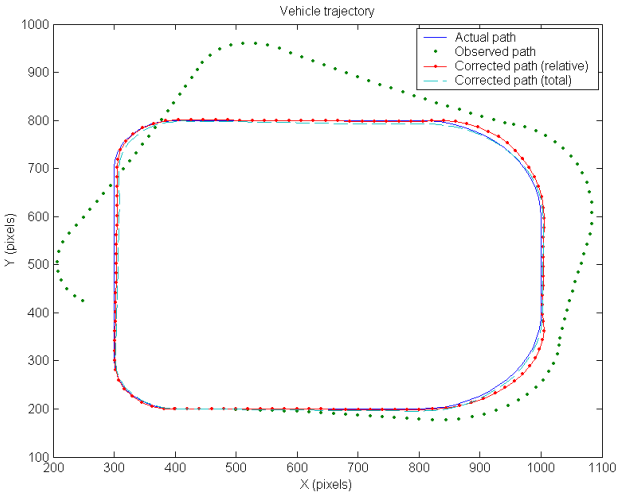

Figure 7: Comparison of observed and corrected trajectories with ground truth

Figure 9 shows the outline of a 170m long vehicle guidepath in our indoor laboratory mapped using pose estimates only from dead-reckoning. The guidepath was mapped with our test platform using a total of 1836 images, and has 4 intersections that are unresolved. Figure 10 shows the resulting map after loop closure was enforced, with the overlap regions highlighted.

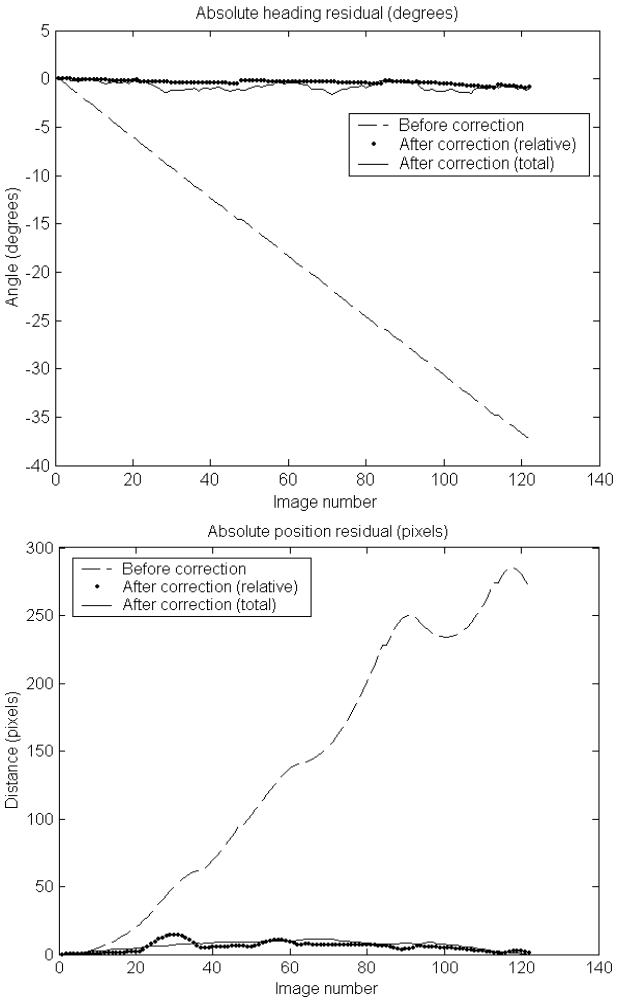

Figure 8: Comparison of residual in absolute heading and position

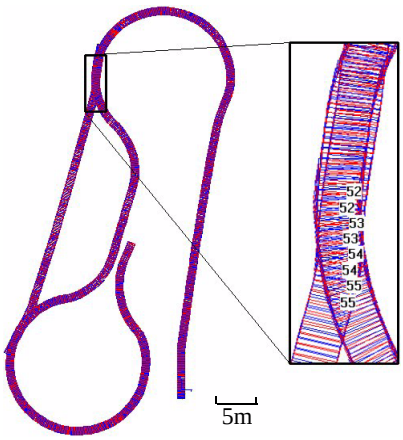

Figure 9: Map built using only initial pose estimates from odometry. Observe absence of cycle-closure at the two lower right locations, including discrepancy in overlap of the numbered features at the top left intersection.

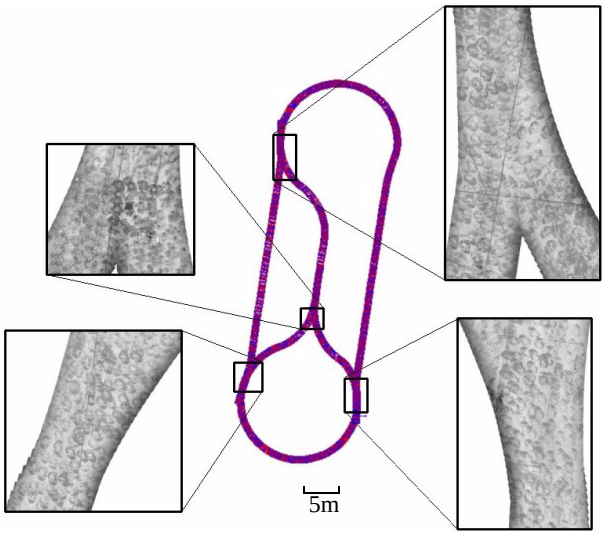

Figure 10: Map after loop-closure. Regions of the final mosaic at the intersections are highlighted.

6 Possible Extensions and Conclusions

It is feasible to extend the work presented to an online algorithm. Along the lines of the Local Registration Global Correlation (LRGC) algorithm [6], images are registered within the K-neighborhood of the current in constant time. On revisiting an area, the algorithm is run using only the constraint equation corresponding to the newly added cycle in the topology graph, using the state covariance propagated from the last such iteration.

What we have presented is an algorithm that resolves inconsistencies in mosaics of cyclic environments in $O(n_l^2 n_i)$ time, where commonly the number of loops $n_l \ll n_i$. Local continuity, which is more crucial to the task of high-speed tracking than global consistency, is also shown to be preserved to a significant extent. In our work, we have generated reliable mosaics ranging from 80m to 800m in length, in laboratory as well as real factory environments. They have been used by our test vehicles at speeds exceeding normal operating conditions for up to 40hr stretches without error requiring human intervention.

The sensitivity of the mapping algorithm to false positives is currently being addressed using reliable techniques for registering images at intersections with sub-pixel accuracy. We also use a diagonally weighted form of the right pseudoinverse solution to compute pose perturbation in a unit that ensures the same amount of distortion at pixels on the boundary of an image for a given small increment in position or orientation. Development of a suitable uncertainty model accounting for the trajectory-dependent nature of odometry error, extensions to mosaicing 2-dimensional areas, and the incorporation of online auto-calibration techniques are also being pursued.

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

[3] 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.

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

[5] 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

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

[7] 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

[8] 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*.

[9] Kelly A., "Pose Determination and Tracking in Image Mosaic-Based Vehicle Position Estimation", *International Conference on Intelligent Robots and Systems (IROS00)*, October, 2000.

[10] Kelly A., "Mobile Robot Localization from Large Scale Appearance Mosaics", tech. report CMU-RI-TR-00-21, Robotics Institute, Carnegie Mellon University, December, 2000

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

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

[13] 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

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

[15] Sawney H.S., Hsu S. and Kumar R., "Robust video mosaicing 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

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

[17] 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.

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

[19] 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