Visual Odometry System Using Multiple Stereo Cameras and Inertial Measurement Unit

Taragay Oskiper Zhiwei Zhu Supun Samarasekera Rakesh Kumar
Sarnoff Corporation, 201 Washington Road, Princeton, NJ 08540, USA
{toskiper, zzhu, ssamarasekera, rkumar}@sarnoff.com

Abstract

Over the past decade, tremendous amount of research activity has focused around the problem of localization in GPS denied environments. Challenges with localization are highlighted in human wearable systems where the operator can freely move through both indoors and outdoors. In this paper, we present a robust method that addresses these challenges using a human wearable system with two pairs of backward and forward looking stereo cameras together with an inertial measurement unit (IMU). This algorithm can run in real-time with 15Hz update rate on a dual-core 2GHz laptop PC and it is designed to be a highly accurate local (relative) pose estimation mechanism acting as the front-end to a Simultaneous Localization and Mapping (SLAM) type method capable of global corrections through landmark matching. Extensive tests of our prototype system so far, reveal that without any global landmark matching, we achieve between 0.5% *and* 1% *accuracy in localizing a person over a 500 meter travel indoors and outdoors. To our knowledge, such performance results with a real time system have not been reported before.*

1. Introduction

Localization in GPS denied environments is a challenging problem which is getting ever increasing amount of attention from researchers in many different fields. The reason which makes this a very appealing problem is because such a technology has a wide variety of potential applications, from autonomous robot/vehicle navigation to tracking soldiers and emergency personnel during training exercises or in active missions. Most of the work in this area so far has focused on active sensing devices such as sonar and laser range finders. However, recently inertial based passive sensors coupled with cameras has attracted increased attention thanks to the ever increasing processor speeds and the advent of compact systems with multiple processors that meet the intensive computational requirements of such configurations. Previous published methods for visual odometry, without trying to be all inclusive as the list of publications in this field is very large, have used video streams from 1 or 2 moving cameras in monocular [3],[4],[10],[13] or binocular [9],[11],[12] configurations. Recent work on invariant feature matching has also lead to landmark based 3D motion tracking systems [14],[15]. However, these systems while impressive are still not robust enough for autonomous use over large distances and time periods. The reported accuracies in localization are between 1% and 5% over distances of few hundred meters, mostly outdoors, (much shorter distances for indoors only).

Figure 1. Front and back views of the backpack system and sample images captured as the person enters a room.

Our multi-camera system is built on the real-time visual odometry algorithm for a single stereo camera developed by [11] and it is implemented with a complete parallel architecture, so that real-time processing can be achieved on a multi-cpu system, where computationally most extensive single camera related routines remain the same and can be carried out by separate CPUs.

We have performed many experiments where we compared visual odometry on a single stereo platform against ground truth data obtained with survey quality differential GPS equipment and it has proved to be extremely accurate as long as the working environment is relatively benign. However if there are abrupt movements that jolt the cameras violently, the field of view contains large moving

1-4244-1180-7/07/$25.00 ©2007 IEEE

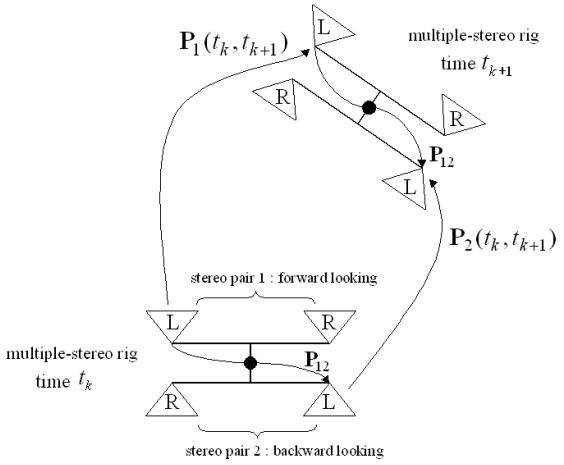

Figure 2. Rig with two stereo pairs moving in 3-D space between time instants t_k and t_{k+1}. Left and right cameras are denoted by letters 'L' and 'R' in each pair. In the multicamera visual odometry algorithm, given $\mathbf{P}_{12}$, the fixed relative pose between the two left cameras of the front and back pairs, and $\mathbf{P}_1(t_k, t_{k+1})$, the relative pose of the left camera of the front pair between two time instants, we can determine $\mathbf{P}_2(t_k, t_{k+1})$ and vice versa. On the other hand, for the automatic calibration procedure, given many pairs of highly accurate poses $\mathbf{P}_1(t_k, t_{k+1})$ and $\mathbf{P}_2(t_k, t_{k+1})$ for $0 \leqq k \leqq N$, one can determine the $\mathbf{P}_{12}$ which makes these pose pairs agree the most according to some criteria.

objects, the cameras come very close to walls or other obstacles, or the images do not contain good features that can be tracked accurately, the pose estimates rapidly deteriorate. These challenging conditions are more common in indoors environments where the user can come close to plain walls with little features or walk down crowded corridors, and while opening doors, in which case the moving door occupies the entire view in the front camera. In order to increase the robustness of the system under such circumstances, we have integrated an additional pair of stereo-cameras facing backward and an IMU unit. The idea being that, whenever the scenery is challenging for one stereo pair, the other pair is utilized and vice versa. Then the problem boils down to deciding which one of the pairs is providing the best pose solution at any given time. If both stereo pairs fail due to the lack of features or bad illumination, then the measurements of the IMU help stabilize the pose estimates.

2. Multicamera Pose Transfer Mechanism

In our system with front and backward facing two stereo pairs, visual odometry can be applied to each one individually to estimate left camera pose in that pair. In each stereo head the extrinsic relation between the left and right cameras is known through calibration. In addition, the pose relation between the front left and back left cameras is also computed through an automatic calibration mechanism which will be described in Section 4. This fixed relation across the two stereo pairs constrains the results and taking advantage of this constraint requires the ability to compute the pose of the left camera in one of the pairs given the pose of the left camera in the other. Below we will briefly describe how this pose transfer method works.

The stereo visual odometry algorithm applied in both front and back cameras separately, provides us with local pose information of the left cameras in each pair between time instants t_k and t_{k+1}. These are denoted by $\mathbf{P}_1(t_k, t_{k+1})$ and $\mathbf{P}_2(t_k, t_{k+1})$ for the front and back pairs respectively. That is to say, the pose of the left camera of the front stereo pair (denoted with subscript 1) at time t_{k+1} in the local coordinate frame of the left camera at time t_k is specified by a rotation matrix $\mathbf{R}_1(t_k, t_{k+1})$ and a translation vector $\mathbf{T}_1(t_k, t_{k+1})$ that transforms the coordinates of a camera point $\mathbf{X}_1(t_k)$ expressed in the camera coordinate frame at time t_k to the camera point $\mathbf{X}_1(t_{k+1})$ expressed in the coordinate frame at time t_{k+1}:

$$\mathbf{X}_1(t_{k+1}) = \mathbf{R}_1(t_k, t_{k+1})\mathbf{X}_1(t_k) + \mathbf{T}_1(t_k, t_{k+1}) \; .$$

This transformation can also be expressed as

$$\begin{bmatrix} \mathbf{X}_1(t_{k+1}) \\ 1 \end{bmatrix} = \mathbf{P}_1(t_k, t_{k+1}) \begin{bmatrix} \mathbf{X}_1(t_k) \\ 1 \end{bmatrix}$$

where

$$\mathbf{P}_1(t_k, t_{k+1}) = \begin{bmatrix} \mathbf{R}_1(t_k, t_{k+1}) & \mathbf{T}_1(t_k, t_{k+1}) \\ 0 & 1 \end{bmatrix} \; .$$

Also, using the extrinsics between the two stereo pairs, the pose of camera 2 (left camera in the back stereo pair) relative to camera 1 (left camera in front stereo pair) is described by $\mathbf{P}_{12}$ such that

$$\mathbf{X}_2(t) = \mathbf{P}_{12}\mathbf{X}_1(t) \; , \; \forall t \text{ with } \mathbf{P}_{12} = \begin{bmatrix} \mathbf{R}_{12} & \mathbf{T}_{12} \\ 0 & 1 \end{bmatrix} \; .$$

Then, the following expressions can be shown to hold:

$$\begin{aligned} \mathbf{P}_2(t_k, t_{k+1}) &= \mathbf{P}_{12}\,\mathbf{P}_1(t_k, t_{k+1})\mathbf{P}_{12}^{-1} \\ \mathbf{P}_1(t_k, t_{k+1}) &= \mathbf{P}_{12}^{-1}\,\mathbf{P}_2(t_k, t_{k+1})\mathbf{P}_{12} \end{aligned} \tag{1}$$

which enables the computation of $\mathbf{P}_2(t_k, t_{k+1})$ given $\mathbf{P}_{12}$ and $\mathbf{P}_1(t_k, t_{k+1})$, and vice versa. Hence, using the fixed extrinsic $\mathbf{P}_{12}$, camera poses in the front stereo pair's local left camera coordinate frame can be transferred to the back camera's local left camera coordinate frame and vice versa. This relation between camera poses on a fixed configuration allows us to determine the poses of all the cameras constrained by any given single camera pose. This way, we can evaluate the quality of poses generated by one stereo pair on the other stereo pair's dataset, the goal being to decide

on the pose that performs well on both datasets. As a result, all the pose candidates get evaluated on the same data including points from all cameras enabling the selection of the best pose to be very robust.

3. Multicamera Visual Odometry Algorithm

Visual odometry addresses the problem of estimating camera poses based on image sequences. After acquiring the left and right camera image frames at time t_k, the first step consists of detecting and matching Harris corner [5] based feature points in each stereo pair. Feature point image coordinates are normalized using the known intrinsic calibration parameters in each camera (through multiplication with the inverse of the calibration matrix) and compensated for radial distortion. In stereo matching process, calibration information allows us to eliminate most of the false matches by applying epipolar and disparity constraints. This is followed by computation of the 3D locations corresponding to these feature points through stereo triangulation in the coordinate frame of the current left camera. Next, using the new image frames at time step t_{k+1}, 2D-2D correspondences are established by matching feature points between the previous frames at timestep t_k and the current ones at t_{k+1}. This allows 3D-2D point correspondences to be established based on the 3D point cloud computed in the previous step. Finally, the pose of the left camera in each stereo pair can be computed using a robust resection method based on RANSAC followed by iterative refinement of the winning hypothesis where Cauchy based robust cost function of the reprojection errors in both the left and right images is minimized. For the front stereo pair ($j = 1$) and back stereo pair ($j = 2$), this cost function is given by

$$c_j(\mathbf{P}_k^j) = \sum_{i=1}^{K_j} \rho(\mathbf{x}_i^{\ell_j} - h(\mathbf{P}_k^j \mathbf{X}_i^j)) + \rho(\mathbf{x}_i^{r_j} - h(\mathbf{P}^{s_j} \mathbf{P}_k^j \mathbf{X}_i^j)) \,, \tag{2}$$

where, for the jth stereo pair, K_j is the number of feature points, $\mathbf{x}_i^{\ell_j}$ and $\mathbf{x}_i^{r_j}$ denote coordinates of the feature point i in the left and right images, $\mathbf{X}_i^j$ denotes its 3D position in homogeneous coordinates, $\mathbf{P}^{s_j}$ denotes the pose of the right camera in the left camera coordinate frame (known through stereo calibration), function h is used in denoting the conversion from homogeneous to inhomogeneous coordinates, $\rho(\mathbf{y}) = \log(1 + \|\mathbf{y}\|^2/a^2)$ is the Cauchy based robust cost function with a given scale parameter a, and finally $\mathbf{P}_k^j \triangleq \mathbf{P}_j(t_k, t_{k+1})$ is the relative pose across two time instants.

In our baseline approach all the above steps are performed independently for both the front and back stereo pairs in a parallel fashion. At the end of this process, two pose estimates are obtained from both pairs. The winning pose out of these two candidates is determined by computing their reprojection errors in the entire system using the pose transfer mechanism (1). Hence, by defining the following global cost functions,

$$\begin{aligned} \bar{c}_1(\mathbf{P}_k^1) &= c_1(\mathbf{P}_k^1) + c_2(\mathbf{P}_{12}\mathbf{P}_k^1\mathbf{P}_{21}) \\ \bar{c}_2(\mathbf{P}_k^2) &= c_2(\mathbf{P}_k^2) + c_1(\mathbf{P}_{21}\mathbf{P}_k^2\mathbf{P}_{12}) \,, \end{aligned} \tag{3}$$

for the front and back pairs respectively, the final decision function used in selecting the best camera index can be written as

$$d(\mathbf{P}_k^1, \mathbf{P}_k^2) = \begin{cases} 1 & \text{if } \bar{c}_1(\mathbf{P}_k^1) < \bar{c}_2(\mathbf{P}_k^2) \\ 2 & \text{otherwise} \end{cases} \tag{4}$$

where we compare cumulative (global) scores, $\bar{c}_1(\mathbf{P}_k^1)$ and $\bar{c}_2(\mathbf{P}_k^2)$, of the camera poses determined by each stereo pair, by which we mean combined scores after that pose is transformed and scored on every camera's data in the multicamera system. For instance, in case where the front camera feature tracking is successful but the back camera is not, the pose generated based on the front camera's data points will have a small residual error in its own data set and a large residual error in the back camera's dataset. On the other hand the pose generated by the back stereo pair will have high residual error not only on its own dataset, but it will also have high residual error on the front camera's dataset as well, since it was computed from poor data to begin with. So this method provides a very robust way of choosing the best pose out of the multiple candidates.

Although in this paper, we present results from this baseline system, some more advanced techniques can be implemented where the two pairs are more intricately integrated resulting in a tightly coupled system. These efforts are to increase not only the robustness of the algorithm but also the overall accuracy over that of either one of the single stereo pairs can achieve. In the following, we describe these in more detail.

3.1. Multi-camera Preemptive RANSAC and Iterative Refinement

In the preemptive RANSAC algorithm for single stereo visual odometry, randomly selected 3 point correspondences between the 3D point cloud and the 2D image points, N number of pose hypotheses (by default N=500) are generated using the 3-point resection algorithm. Then, all the hypotheses are evaluated on chunks of M data points (by default M=100) based on the robust cost function (2) , each time dropping out half of the least scoring hypotheses. Thus, initially we start with 500 hypotheses, all of which are evaluated on a subset of 100-point correspondences. Then they are sorted according to their scores on this data set and the bottom half is removed. In the next step, another set of 100 data points is selected on which the remaining 250 hypotheses are evaluated and the least scoring half are pruned and so forth until a single best-scoring hypothesis remains.

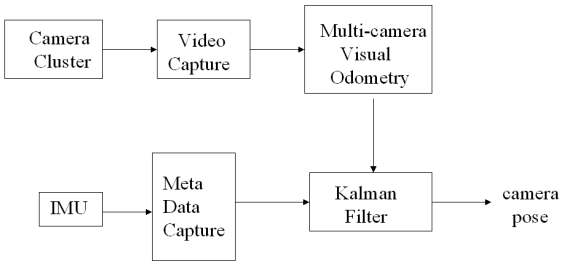

Figure 3. Flow diagram for multicamera visual odometry and IMU integration in our system.

For the multi-camera system, preemptive RANSAC can be implemented as follows. As before, each camera independently generates 500 pose hypotheses that are based on randomly selected 3 point correspondences using its own data. However, these hypotheses from each camera are evaluated not only on the camera that generated them but their transformations are also evaluated on the other stereo pair's data using the global cost functions (3). Hence, preemptive scoring in each camera is accomplished by obtaining a cumulative score for each hypothesis after combining its corresponding scores determined in all the cameras on the initial set of 100 data points from each camera. Then the least scoring half (now based on cumulative score) from each camera is discarded and the remaining half is evaluated on another set of 100 points in every camera and the cumulative scores are updated and so forth. So at the end of the preemptive RANSAC the best cumulative scoring hypothesis is obtained in each camera.

At the end of the preemptive RANSAC routine, the winning hypothesis in each stereo pair is passed to a pose refinement (polishing) step where iterative minimization of the robust cost function of the reprojection errors is performed through Levenberg-Marquardt method. For this polishing step in the single stereo camera case, the cost function in (2) is used, however for the multicamera case the cumulative cost functions given by (3) can be used in each stereo pair resulting in a better pose estimate than either of them could produce on its own.

4. Automatic Extrinsic Calibration

We perform stereo calibration (intrinsic and extrinsic) for each pair using the Matlab calibration toolkit [1]. Also, as described in Section 2, for the multi-camera visual odometry algorithm we assume that $\mathbf{P}_{12}$ is known beforehand which is used in the algorithm to constrain the pose of one camera pair given the other.

In order to determine $\mathbf{P}_{12}$, one option would be to collect several images from all cameras by placing the rig in a calibration room with known 3-D reference points. Then the parameters of the extrinsic pose relation between the front left and back left cameras would be found using an optimization tool that would minimize a suitable function of the reprojection errors in the images. Instead of such an approach, we chose to develop a convenient automatic calibration method which served well. The idea behind this method is based on the fact that given the left camera poses in the two stereo pairs, one can determine $\mathbf{P}_{12}$. Namely, if we ran visual odometry on both pairs independently in a feature rich environment which would make sure both systems perform equally well and provide highly accurate poses, then the outputs from both pairs could be used in order to search for the extrinsic relation that would bring these two sources into agreement according to some criterion described below. We collected such calibration quality data outdoors using a feature rich environment in brightly lit conditions where the person wearing the backpack system walked slowly on a long circular path while at the same time revolving around his body, making many 360 degree turns along the way. This kind of data proved to be very useful as it exercised all degrees of freedom simultaneously. This problem is related to the sequence alignment work in [2] where the goal is to solve for the homography between two cameras based on the non-overlapping image sequences obtained from both.

Hence, given a sequence of pose estimates for $0 \leqq k \leqq N$, where N is in the order of several thousand, in order to determine $\mathbf{P}_{12}$, we start by writing

$$\mathbf{P}_{12} = \begin{bmatrix} \mathbf{R}(\mathbf{q}) & \mathbf{T} \\ 0 & 1 \end{bmatrix}$$

in terms of the unknown quantities $\mathbf{T} \triangleq \mathbf{T}_{12}$ (a 3-vector) and $\mathbf{q}$, the unit quaternion (4-vector) used for parametrization of the rotation matrix $\mathbf{R}_{12}$ such that

$$\mathbf{R}(\mathbf{q}) = (|q_w|^2 - \|\overrightarrow{\mathbf{q}}\|^2)\mathbf{I}_{3\times 3} + 2\overrightarrow{\mathbf{q}}\,\overrightarrow{\mathbf{q}}^T + 2q_w \left[\overrightarrow{\mathbf{q}}\right]_\times$$

where $\mathbf{q} = \left[\overrightarrow{\mathbf{q}}^T \; q_w\right]^T$ and $\overrightarrow{\mathbf{q}} = [q_x \; q_y \; q_z]^T$ and $\left[\overrightarrow{\mathbf{q}}\right]_\times$ denotes the skew-symmetric matrix formed by the components of the vector $\overrightarrow{\mathbf{q}}$, [7],[8].

Then using (1)

$$\mathbf{P}_{12}\mathbf{P}_1(t_k, t_{k+1}) = \mathbf{P}_2(t_k, t_{k+1})\mathbf{P}_{12}$$

we obtain the following expressions

$$\mathbf{R}(\mathbf{q})\mathbf{R}_1(k) = \mathbf{R}_2(k)\mathbf{R}(\mathbf{q}) \tag{5}$$

$$\mathbf{R}(\mathbf{q})\mathbf{T}_1(k) + \mathbf{T} = \mathbf{R}_2(k)\mathbf{T} + \mathbf{T}_2(k) \tag{6}$$

where we used $\mathbf{R}_1(k) \triangleq \mathbf{R}_1(t_k, t_{k+1})$ and $\mathbf{T}_1(k) \triangleq \mathbf{T}_1(t_k, t_{k+1})$ for ease of notation (similarly for camera 2).

However, since these pose estimates will be erroneous, no matter how small the errors are, the equalities in (5)

and (6) will not be perfectly satisfied. Instead, based on these quantities we choose a suitable objective function, $\sum_{k=0}^{N}\|\boldsymbol{\epsilon}_k\|^2$, that we seek to minimize:

$$\begin{aligned}\boldsymbol{\epsilon}_k =&\big(\mathbf{R}_2(k)\mathbf{R}(\mathbf{q})\big)^T\big(\mathbf{R}_2(k)\mathbf{T}+\mathbf{T}_2(k)\big)\\ &-\big(\mathbf{R}(\mathbf{q})\mathbf{R}_1(k)\big)^T\big(\mathbf{R}(\mathbf{q})\mathbf{T}_1(k)+\mathbf{T}\big)\\ =&\mathbf{R}(\mathbf{q})^T\mathbf{R}_2^T(k)\,\mathbf{T}_2(k)+\big(\mathbf{R}_1^T(k)-\mathbf{I}_{3\times3}\big)\,\mathbf{t}\\ &-\mathbf{R}_1^T(k)\mathbf{T}_1(k)\end{aligned} \tag{7}$$

where $\mathbf{t}=-\mathbf{R}(\mathbf{q})^T\mathbf{T}$. This error function has the following interpretation: We seek for the pose relation $\mathbf{P}_{12}$ that minimizes the sum of the frame to frame differences in the position of the left camera center of the back stereo pair at time t_{k+1} in the coordinate frame of the left camera of the front stereo pair at time t_k, which are computed by using visual odometry in the front and back pairs separately.

To solve for the unknown parameters, $\mathbf{t}$ and $\mathbf{q}$, we use RANSAC method followed by Levenberg-Marquardt based iterative minimization on the inlier set. During the RANSAC process, based on the relation in (5) we randomly select 3 rotation matrix pairs $\mathbf{R}_1(k_i)$, $\mathbf{R}_2(k_i)$, $1\leqq i\leqq 3$, and solve for the unit norm $\mathbf{q}$ that minimizes

$$\sum_{i=1}^{3}\|\mathbf{q}_1(k_i)\otimes\mathbf{q}-\mathbf{q}\otimes\mathbf{q}_2(k_i)\|^2 \tag{8}$$

where $\otimes$ represents quaternion product and $\mathbf{q}_1(k_i)$ and $\mathbf{q}_2(k_i)$ are the quaternion representations corresponding to $\mathbf{R}_1(k_i)$ and $\mathbf{R}_2(k_i)$ respectively [7], [8]. The above expression can be written as

$$\min_{\substack{\mathbf{q}\\ \text{s.t.}\|\mathbf{q}\|=1}}\sum_{i=1}^{3}\big\|\big(\mathbf{Q}_1(k_i)-\mathbf{Q}_2(k_i)\big)\mathbf{q}\big\|^2 \tag{9}$$

where $\mathbf{Q}_1(k_i)$ and $\mathbf{Q}_2(k_i)$ are 4×4 matrices formed by the elements of $\mathbf{q}_1(k_i)$ and $\mathbf{q}_2(k_i)$, used in the matrix representation of quaternion product [8]. Letting $\mathbf{C}_i\triangleq\mathbf{Q}_1(k_i)-\mathbf{Q}_2(k_i)$ and $\mathbf{C}=[\mathbf{C}_1^T\,\mathbf{C}_2^T\,\mathbf{C}_3^T]^T$, (9) can be simplified as $\|\mathbf{C}\mathbf{q}\|^2$ where the solution $\mathbf{q}$ is obtained as the unit eigenvector corresponding to the smallest eigenvalue of the matrix $\mathbf{C}^T\mathbf{C}$.

Next, letting $\hat{\mathbf{q}}$ be the solution obtained above and based on relation (6), we solve for the $\mathbf{T}$ which minimizes

$$\sum_{i=1}^{3}\|\big(\mathbf{I}_{3\times3}-\mathbf{R}_2(k_i)\big)\mathbf{T}+\mathbf{R}(\hat{\mathbf{q}})\mathbf{T}_1(k_i)-\mathbf{T}_2(k_i)\|^2\,. \tag{10}$$

Letting $\mathbf{A}_i=\mathbf{I}_{3\times3}-\mathbf{R}_2(k_i)$, $\mathbf{b}_i=\mathbf{R}(\hat{\mathbf{q}})\mathbf{T}_1(k_i)-\mathbf{T}_2(k_i)$, $\mathbf{A}=[\mathbf{A}_1^T\,\mathbf{A}_2^T\,\mathbf{A}_3^T]^T$, and $\mathbf{b}=[\mathbf{b}_1^T\,\mathbf{b}_2^T\,\mathbf{b}_3^T]^T$, (10) simplifies to, $\min_{\mathbf{T}}\|\mathbf{A}\mathbf{T}-\mathbf{b}\|^2$ from which $\mathbf{T}$ can be obtained.

The best hypothesis determined by the RANSAC method is then refined in an iterative process. At iteration step j, using the small rotation quaternion $\delta\mathbf{q}\approx[\overrightarrow{\delta\mathbf{q}}\quad 1]^T$ such that $\mathbf{R}(\delta\mathbf{q})\approx\mathbf{I}_{3\times3}+2\left[\overrightarrow{\delta\mathbf{q}}\right]_\times$, for the correction step in the rotation matrix and the translation vector, $\mathbf{R}(\mathbf{q}^{(j+1)})$=$\mathbf{R}(\delta\mathbf{q})\mathbf{R}(\mathbf{q}^{(j)})$ and $\mathbf{t}^{(j+1)}=\mathbf{t}^{(j)}+\delta\mathbf{t}$, the jacobian $\mathbf{J}_k$ of the error function $\boldsymbol{\epsilon}_k$ in the parameter set $\{\delta\mathbf{t},\overrightarrow{\delta\mathbf{q}}\}$ can be shown to be

$$\mathbf{J}_k=\Big[\mathbf{R}_1^T(k)-\mathbf{I}_{3\times3}\;;\;2\mathbf{R}(\mathbf{q})^T\left[\mathbf{R}_2^T(k)\mathbf{T}_2(k)\right]_\times\Big]$$

which is used in the Levenberg-Marquardt algorithm.

5. Visual Odometry and IMU integration

Even with a second stereo pair, there are still situations (although greatly minimized) where both pairs provide poor pose outputs. For instance, during a turn in a staircase where both cameras face the surrounding white walls for a brief moment. So, in order to further increase the robustness, we integrated our visual odometry system with a MEMS (Microelectromechanical Systems) based IMU using the extended Kalman filter (EKF) framework. Similar to [3], we chose a " constant velocity, constant angular velocity" model for the filter dynamics. The state vector consists of 16 elements: $\mathbf{X}$, 3-vector representing position in navigation coordinates, $\mathbf{q}$, unit quaternion (4-vector) for attitude representation in navigation coordinates, $\mathbf{v}$, 3-vector for translational velocity in body coordinates, $\boldsymbol{\omega}$, 3-vector for rotational velocity in body coordinates, and $\mathbf{b}$, 3-vector for gyro bias components in all three rotational axes. We choose quaternion representation for attitude since it has several practical properties. Each component of the rotation matrix in quaternion is algebraic, eliminating the need for transcendental functions. It is also free of the singularities that are present with other representations and the prediction equations are treated linearly. Based on this, the process model we adopted is given by

$$\begin{aligned}\mathbf{X}_k&=\mathbf{X}_{k-1}+\mathbf{R}^T(\mathbf{q}_{k-1})\mathbf{x}_{rel}\\ \mathbf{q}_k&=\mathbf{q}_{k-1}\otimes\mathbf{q}(\boldsymbol{\rho}_{rel})\\ \boldsymbol{\omega}_k&=\boldsymbol{\omega}_{k-1}+\mathbf{n}_{w,k-1}\\ \mathbf{b}_k&=\mathbf{b}_{k-1}+\mathbf{n}_{b,k-1}\\ \mathbf{v}_k&=\mathbf{v}_{k-1}+\mathbf{n}_{v,k-1}\end{aligned} \tag{11}$$

where

$$\begin{aligned}\mathbf{x}_{rel}&=\mathbf{v}_{k-1}\Delta t_k+\mathbf{n}_{v,k-1}\Delta t_k\\ \boldsymbol{\rho}_{rel}&=\boldsymbol{\omega}_{k-1}\Delta t_k+\mathbf{n}_{w,k-1}\Delta t_k\end{aligned} \tag{12}$$

and $\otimes$ is used to denote the quaternion product operation. Above, $\boldsymbol{\rho}_{rel}$ is the rotation vector (representing the rotation axis) in the body frame, $\mathbf{R}(\mathbf{q})$ is the rotation matrix determined by the attitude quaternion $\mathbf{q}$ in the navigation frame, and $\mathbf{q}(\boldsymbol{\rho}_{rel})$ is the quaternion obtained from the rotation

vector $\boldsymbol{\rho}$. White Gaussian noise $\{\mathbf{n}_{b,k}\}$ is used in the gyro bias process model. Undetermined accelerations in both translational and angular velocity components are modeled by zero mean white Gaussian noise processes $\{\mathbf{n}_{v,k}\}$ and $\{\mathbf{n}_{w,k}\}$. The filter runs at the frame rate, meaning that the discrete time index denoted by k corresponds to the frame times when pose outputs are also available from visual odometry system.

The multicamera visual odometry frame to frame local pose measurements expressed in the coordinate frame of the front left camera, $\mathbf{P}_k \triangleq \mathbf{P}(t_k, t_k+1)$, are converted to velocities by extracting the rotation axis vector corresponding to the rotation matrix $\mathbf{R}_k$, together with the camera translation given by $-\mathbf{R}^T\mathbf{T}_k$, (where $\mathbf{P}_k = [\mathbf{R}_k;\ \mathbf{T}_k]$) and then dividing by the timestep, $\Delta t_k = t_{k+1} - t_k$. The angular rate sensor (gyro) and accelerometer readings from the IMU are used directly as measurements in the Kalman filter. Hence, the observations from visual odometry and IMU are used according to the following measurement model:

$$\begin{aligned} \mathbf{v}_k^{\text{vo}} &= \mathbf{v}_k + \mathbf{n}_{v,k}^{\text{vo}} \\ \boldsymbol{\omega}_k^{\text{vo}} &= \boldsymbol{\omega}_k + \mathbf{n}_{w,k}^{\text{vo}} \\ \boldsymbol{\omega}_k^{\text{imu}} &= \boldsymbol{\omega}_k + \mathbf{b}_k + \mathbf{n}_{w,k}^{\text{imu}} \\ \mathbf{a}_k^{\text{imu}} &= \mathbf{R}(\mathbf{q}_k)\,\mathbf{g} + \mathbf{n}_{a,k}^{\text{imu}}\,. \end{aligned} \tag{13}$$

Here, $\mathbf{v}_k^{\text{vo}}$ and $\boldsymbol{\omega}_k^{\text{vo}}$ are translational and angular velocity measurements provided by visual odometry (vo), $\boldsymbol{\omega}_k^{\text{imu}}$ and $\mathbf{a}_k^{\text{imu}}$ are the gyro and accelerometer outputs provided by the IMU and $\mathbf{g} = [0\ 0\ 9.8]^T$ is the gravity vector . Uncertainty in the visual odometry pose estimates, represented by the noise components $\mathbf{n}_{v,k}^{\text{vo}}$ and $\mathbf{n}_{\omega,k}^{\text{vo}}$, is estimated based on the reprojection error covariance of image features through backward propagation [6]. The gyro noise errors are modeled with fixed standard deviation values that are much higher than those corresponding to the visual odometry noise when the pose estimates are good (which is most often the case) and are comparable in value or sometimes much less when vision based pose estimation is difficult for brief durations. This allows the filter to effectively combine the two measurements at each measurement update, relying more on the sensor with the better noise characteristics. During filter operation bad measurements from all sensors are rejected using validation mechanisms based on Chi-square tests on the Kalman innovations. In addition, those measurements from visual odometry causing large accelerations are also discarded. Also, note that since the uncertainty in the absolute location component $\mathbf{X}_k$ grows without bound in the lack of global position measurements, we perform periodic resets on the filter.

In our first approach, we did not include the gyro bias as part of the filter state. We observed that throughout most of the data sequences , there is a high degree of agreement between the angular velocities computed by visual odometry alone and those available from the gyros. However, if we always used the gyro angular velocity measurements alone, *e.g.*, by removing the second equation from the above measurement model, then we notice that there is a very large drift in the overall trajectory. So, although the difference between the angular velocities at each time instant are small (at those times when visual odometry is working well), the approximation errors in the conversion of velocities to relative poses and gyro bias add up quickly over time. On the other hand, visual odometry trajectory is very accurate except for brief regions where it might "break" causing gross errors in the global sense. And these are the exact moments where we would like to take the most advantage of the IMU measurements. So, in order not to corrupt the visual odometry measurements at all when it is working well, we computed at each time instant the difference in velocities in all three rotation axes provided by both sources and compared to a threshold. If this difference in all axes was smaller than this threshold (which is satisfied more than 90% of the time in all our datasets), then we removed the third equation from the measurement model, meaning that gyro observations were not used. This also served as a double check on the quality of the visual odometry output, in the sense that if it is close to the gyro output we choose the visual odometry alone, which totally eliminates any corruption from the gyro measurements at those time instants. However, when we later tested on a different system configuration and placed the cameras on a car, we noticed that the threshold values computed on the backpack did not work as effectively on the other system. In order to prevent this dependence and the effect of hard thresholds which proved to be not the most robust choice, we included the bias as part of the filter state so that, the time varying gyro bias is always estimated based on the visual odometry outputs, thereby eliminating the need to implement such a threshold mechanism and the need to treat the third equation in the aforementioned manner.

6. Results

Our current prototype system consists of 4 digital cameras producing gray-scale images at 640x480 resolution. All the cameras and the IMU unit are externally triggered to provide very accurate synchronization across all components. We have performed many tests where the person wearing the system walks in indoor and outdoor environments along a predefined path. Here, we will present some sample results to demonstrate its current localization performance.

For the auto calibration procedure, we collected a sequence for which we had 2411 pose estimates obtained with both stereo pairs. Applying the algorithm described in Section 4, we obtained $\mathbf{T} = [0.3345\ \ 0.0667\ \ -0.1138]$ in meters and the roll, pitch, yaw angles as -0.9267, -30.2715, -179.6150 degrees respectively, based on the inlier set of

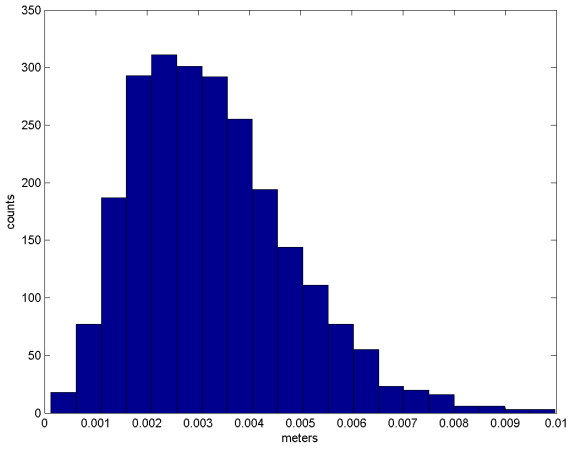

Figure 4. Histogram of the sequence alignment errors $\|\boldsymbol{\epsilon}_k\|, 0 \leqq k \leqq 2392$, based on the pose solution $\mathbf{P}_{12}$ obtained by the auto-calibration algorithm.

2392 data points. In Figure 4, the histogram of the alignment errors, $\|\boldsymbol{\epsilon}_k\|, 0 \leqq k \leqq 2392$, is shown. Average error norm in this inlier set is found to be 3.29 millimeters.

In Figure 5, we demonstrate the benefit of IMU and visual odometry integration. In certain situations such as poor illuminations or non-textured scenes, the captured images of the cameras fail to provide sufficient features for the pose estimation so that the visual odometry fails to work properly. For example, as shown in the thumbnail images at the top, there might be cases when all the cameras see mostly white walls so that very few features concentrated in a small portion of the scene are extracted. As a result, the visual odometry alone cannot estimate pose accurately and may cause gross errors or "breaks". The plots at the bottom part, show the angular rate measurements from IMU (blue), visual odometry (red) as well as the filter output (green). Since the designed Kalman filter is capable of producing the optimal output by combining with the best sensor measurement, the filter output closely follows the visual odometry measurements majority of the time when that is the most accurate, and then follows the IMU measurements mostly during these difficult portions of the sequence.

In Figure 6, we show the final trajectory obtained by our system after a 530 meter long outdoor/indoor sequence. To make this visually more clear we manually overlaid it on a map. In comparison, Figure 7 shows the trajectories obtained by the front and back pairs alone, where due to gross errors in several spots both result in large drifts. In this sequence, the person with the backpack enters and exits two buildings, opens and closes several doors along the way.

In Figure 8, we show results from a 264 meter long completely indoor sequence where the person climbs up and down one flight of stairs in two different places and returns to the same spot where he started.

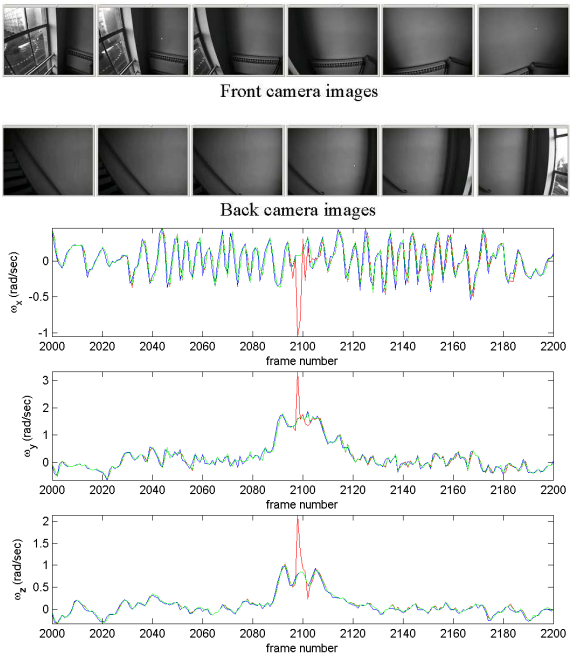

Figure 5. At the top are a sequence of images captured while the person is turning at the bottom of a staircase. They correspond to the section between frames 2096 to 2106. This region of the sequence is challenging for visual odometry as both stereo pairs have very limited scene content. At the bottom, in red are the multicamera visual odometry based angular rates, in blue are the gyro outputs and in green are the Kalman filter outputs. Notice how the Kalman filter output follows the gyro measurements in this brief period, as desired during the failure of visual odometry estimates.

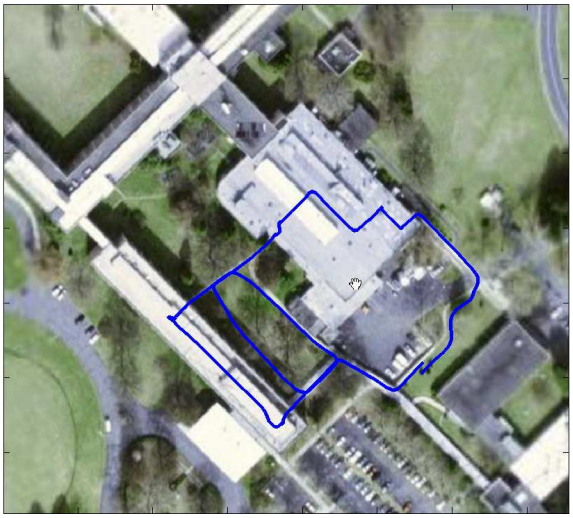

Figure 6. Trajectory obtained by our system from a 530 meter long outdoor/indoor sequence overlaid in 'blue' on a map. Loop closure error is 3.9 meters.

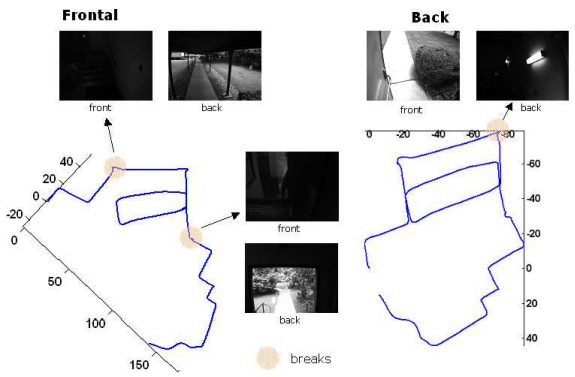

Figure 7. Visual odometry trajectories obtained by the individual stereo pairs alone. The regions where gross errors, "breaks", occur are highlighted and some thumbnail images corresponding to those moments are shown.

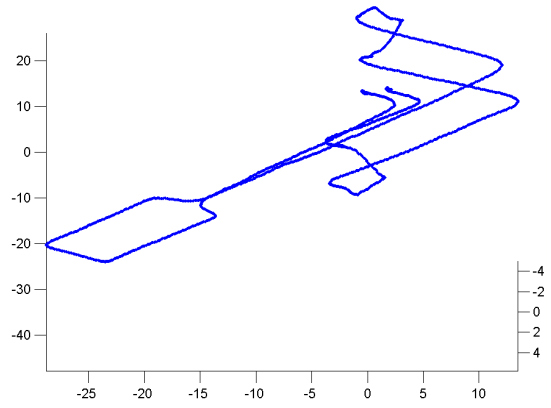

Figure 8. Trajectory obtained by our system from a 264 meter long indoor sequence that includes staircase climbing. Loop closure error is 2.1 meters.

7. Conclusion

In this paper, we described a visual odometry system based on multiple stereo cameras and an inertial measurement unit which we proposed as a powerful method to address the problem of localization in GPS denied environments. We provided results demonstrating the robustness of our approach against many challenges facing such a system and showed that it can provide reliable pose estimates over long distances. For future work, we will continue improving our algorithms, especially concentrating our efforts in the visual odometry and IMU integration framework which is a key component in further increasing the achievable robustness and overall performance in localization accuracy. We also plan to miniaturize our system configuration and test helmet mounted cameras to make it more user friendly.

References

[1] J.-Y. Bouguet. *Camera Calibration Toolbox for Matlab*. www.vision.caltech.edu, 2006.

[2] Y. Caspi and M. Irani. Alignment of non-overlapping sequences. In *IEEE International Conference on Computer Vision*, 2001.

[3] A. Davison. Real-time simultaneous localization and mapping with a single camera. In *IEEE International Conference on Computer Vision*, 2003.

[4] A. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Proc. European Conference on Computer Vision*, 1998.

[5] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, 1988.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[7] B. K. P. Horn. *Robot Vision*. The MIT Press, 1986.

[8] J. B. Kuipers. *Quaternions and Rotation Sequences*. Princeton University Press, 1998.

[9] A. Milella and R. Siegwart. Stereo-based ego-motion estimation using pixel tracking and iterative closest point. In *IEEE International Conference on Computer Vision Systems*, 2006.

[10] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756 – 770, 2004.

[11] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[12] C. Olson, L. Matthies, M. Schoppers, and M. Maimone. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43(4):215–229, 2003.

[13] S. Roumeliotis, A. Johnson, and J. Montgomery. Augmenting inertial navigation with image-based motion estimation. In *IEEE International Conference on Robotics and Automation*, 2002.

[14] D. L. S. Se and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–758, 2002.

[15] D. L. S. Se and J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364–375, 2005.