Large Scale Visual Odometry for Rough Terrain

Kurt Konolige, Motilal Agrawal and Joan Solà
SRI International
333 Ravenswood Ave Menlo Park, CA 94025
konolige,agrawal,sola@ai.sri.com

***Abstract*— Motion estimation from imagery, sometimes called visual odometry, is a well-known process. However, it is difficult to achieve good performance using standard techniques. In this paper, we present the results of several years of work on an integrated system to localize a mobile robot in rough outdoor terrain using visual odometry, with an increasing degree of precision. We discuss issues that are important for real-time, high-precision performance: choice of features, matching strategies, incremental bundle adjustment, and filtering with inertial measurement sensors. Using data with ground truth from an RTK GPS system, we show experimentally that our algorithms can track motion, in off-road terrain, over distances of 10 km, with an error of less than 10 m (0.1%).**

I. Introduction

Estimating motion from an image sensor is an appealing idea - a low-cost, compact and self-contained device based on vision could be an important component of many navigation systems. One can imagine household robots that keep track of where they are, or automatic vehicles that navigate on- and off-road terrain, using vision-based motion estimation to fuse their sensor readings in a coherent way. These tasks are currently entrusted to various types of sensors - GPS and inertial measurement units (IMUs) are the primary ones - that can be costly in high-precision applications, or prone to error: GPS does not work indoors or under tree canopy, low-cost IMUs quickly degrade unless corrected. Visual motion estimation can be a method for estimating motion in is own right, and also as a complement to these more traditional methods.

We are interested in very precise motion estimation over courses that are many kilometers in length, in natural terrain. Vehicle dynamics and outdoor scenery can make the problem of matching images very challenging. Figure 1 shows two successive images from an outdoor scene, with Harris corners extracted. Note the lack of distinctive corner features that are normally present in indoor and urban scenes – most corners in the first image are not found in the second.

In this paper, we present a state-of-the-art system for realtime VO on rough terrain using stereo images (it also works well in indoor and urban scenes). The system derives from recent research by the authors and others on high-precision VO. The most important techniques are:

- Stable feature detection. Precise VO depends on features that can be tracked over longer sequences. Standard features (Harris [11], FAST [19], SIFT [14]) can exhibit poor performance over some offroad sequences. We present a new multiscale feature (named CenSurE [3]) that has improved stability over both outdoor and indoor sequences, and is inexpensive to compute.
- Incremental sparse bundle adjustment. Sparse bundle adjustment (SBA) [9], [24] is a nonlinear batch optimization over camera poses and tracked features. Recent experiments have shown that an incremental form of SBA can reduce the error in VO by a large factor [9], [16], [23].
- IMU integration. Fusing information from an IMU can lower the growth of angular errors in the VO estimate [22]. Even low-cost, low-precision IMUs can significantly increase performance, especially in the tilt and roll axes, where gravity gives a global normal vector.

Our main contribution is a realtime system that, by using these techniques, attains precise localization in rough outdoor terrain. A typical result is less than 0.1% maximum error over a 9 km trajectory, with match failure of just 0.2% of the visual frames, on a vehicle running up to 5 m/s.

II. Visual Odometry Overview

Visual odometry estimates a continuous camera trajectory by examining the changes motion induces on the images. There are two main classes of techniques.

Dense motion algorithms, also known as *optical flow*, track the image motion of brightness patterns over the full image [5]. The computed flow fields are typically useful for obstacle avoidance or other low-level behaviors, and it is difficult to relate flow fields to global geometry. One notable exception is the recent work on stereo flow [6], which uses standard structure from motion (SfM) techniques to relate all the pixels of two successive images. This research is similar to that described in this paper, but uses dense pixel comparisons rather than features to estimate motion.

Feature tracking methods track a small number of features from image to image. The use of features reduces the complexity of dealing with image data and makes realtime performance more realistic. Broadly speaking, there are two approaches to estimating motion. *Causal systems* associate 3D landmarks with features, and track the camera frame with respect to these features, typically in a Kalman filter framework [7], [8], [21]. The filter is composed of the set of landmarks and the last pose of the camera. These methods have not yet proven capable of precise trajectory tracking

This work was performed as part of DARPA's Learning Applied to Ground Robots project.

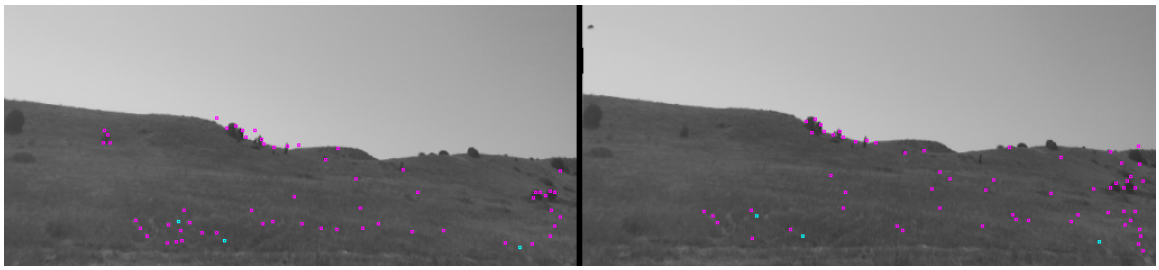

Fig. 1. Harris corner features in two consecutive outdoor frames. Only three matched points survive a motion consistency test.

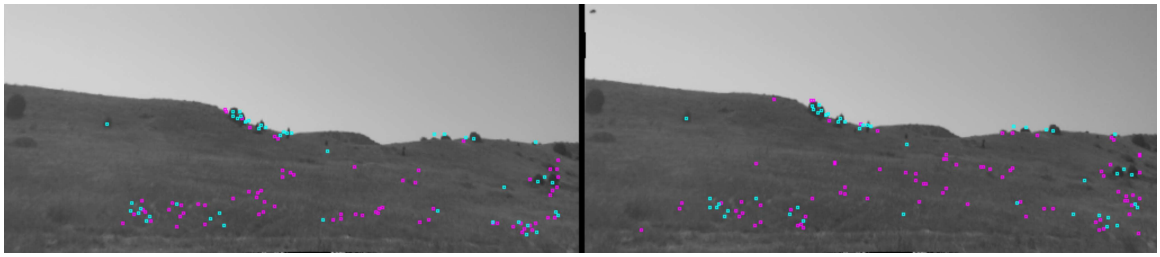

Fig. 2. Matched CenSurE points across two consecutive frames. 94 features are matched, with 44 consensus inliers.

over long distances, because of the computational cost of using large numbers of features in the filter, and because the linearization demanded by the filter can lead to suboptimal estimates for motion.

SfM systems are a type of feature-tracking method that use structure-from-motion methods to estimate the relative position of two or more camera frames, based on matching features [1], [2], [4], [16], [18]. It is well-known that the relative pose (up to scale) of two internally-calibrated camera frames can be estimated from 5 matching features [17], and that two calibrated stereo frames require just 3 points [12]. More matched features lead to greater precision in the estimate - typically hundreds of points can be used to give a high accuracy. The precision of the estimate can be increased by keeping some small number of recent frames in a *bundle adjustment* [9], [16], [23].

Ours is an SfM system, and is most similar to the recent work of Mouragnon et al. [16] and Sunderhauf et al. [23]. The main difference is the precision we obtain by the introduction of a new, more stable feature, and the integration of an IMU to maintain global pose consistency. We also define error statistics for VO in a correct way, and present results of for a vehicle navigating over several kilometers of rough terrain.

A. A Precise Visual Odometry System

Consider the problem of determining the trajectory of a vehicle in unknown outdoor terrain. The vehicle has a stereo camera whose intrinsic parameters and relative pose are known, as well as an IMU with 3-axis accelerometers and gyroscopes. Our goal is to precisely determine the global orientation and position of the vehicle frame at every stereo frame. The system operates incrementally; for each new frame, it does the following.

1) Extract features from the left image.
2) Perform dense stereo to get corresponding positions in the right image.
3) Match to features in previous left image using ZNCC.
4) Form consensus estimate of motion using RANSAC on three points.
5) Bundle adjust most recent N frames.
6) Fuse result with IMU data.

B. Features and Tracking

Distinctive features are extracted from each new frame, then matched to the previous frame by finding the minimum zero-mean normalized cross correlation (ZNCC) score to all features within a search area. ZNCC does not account for viewpoint changes, and more sophisticated methods (affine transform [20], SIFT descriptors [14]) could be used, at the expense of increased computation.

Restricting the search area can increase the probability of a good match. A model of the camera motion (e.g., vehicle dynamics and kinematics for a mounted camera) can predict where to search for features. Let C be the predicted 3D motion of the frame $j+1$ relative to frame j, with covariance Q. If q_i is a 3D point in the coordinates of camera frame j, then its projection $q_{i,j+1}$ and its covariance on camera frame $j+1$ are given by:

$$q_{ij} = KT_C q_i \quad (1)$$

$$Q_{ij} = JQJ^\top. \quad (2)$$

Here K is the camera calibration matrix, T_C is the homogeneous transform derived from C, and J is the Jacobian of q_{ij} with respect to C. In our experiments, we use the predicted angular motion from an IMU, when available, to center the search area using (1), and keep a constant search radius.

From these uncertain matches, we recover a consensus pose estimate using a RANSAC method [10]. Several thousand relative pose hypotheses are generated by randomly se-

lecting three matched non-colinear features, and then scored using pixel reprojection errors (1). If the motion estimate is small and the percentage of inliers is large enough, we discard the frame, since composing such small motions increases error. Figure 3 shows a set of points that are tracked across several key frames.

C. Center Surround Extrema (CenSurE) Features

The biggest difficulty in VO is the data association problem: correctly identifying which features in successive frames are the projection of a common scene point. It is important that the features be *stable* under changes in lighting and viewpoint, *distinctive*, and fast to compute. Typically corner features such as Harris [11] or the more recent FAST [19] features are used. Multiscale features such as SIFT [14] attempt to find the best scale for features, giving even more viewpoint independence. In natural outdoor scenes, corner features can be difficult to find. Figure 1 shows Harris features in a grassy area of the Ft. Carson dataset (see Section III for a description). Note that there are relatively few points that are stable across the images, and the maximum consistent consensus match is only 3 points.

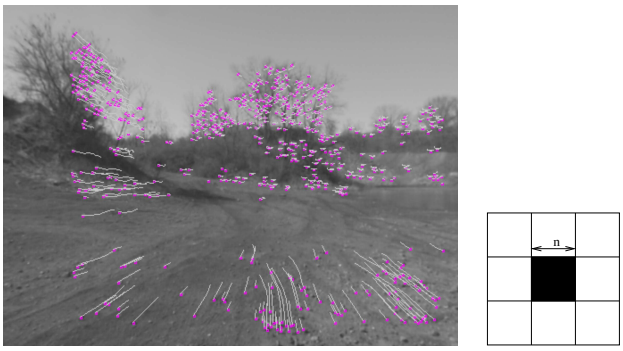

Fig. 3. Left: CenSurE features tracked over several frames. Right: CenSurE kernel of block size n.

The problem seems to be that corner features are small and vanish across scale or variations of texture in outdoor scenes. Instead, we use *center-surround* feature, either a dark area surround by a light one, or vice versa. This feature is given by the normalized Laplacian of Gaussian (LOG) function:

$$\sigma^2 \ \nabla^2 G(\sigma), \tag{3}$$

where $G(\sigma)$ is the Gaussian of the image with a scale of σ. Scale-space extrema of (3) are more stable than Harris or other gradient features [15].

We calculate the LOG approximately using simple center-surround Haar wavelets [13] at different scales. Figure 3(b) shows a generic center-surround wavelet of block size n that approximates LOG; the value $H(x, y)$ is 1 at the light squares, and -8 (to account for the different number of light and dark pixels) at the dark ones. Convolution is done by multiplication and summing, and then normalized by the area of the wavelet:

$$(3n)^{-2} \times \sum_{x,y} H(x, y)I(x, y). \tag{4}$$

which approximates the normalized LOG. These features are very simple to compute using integral image techniques [25], requiring just 7 operations per convolution, regardless of the wavelet size. We use a set of 6 scales, with block size $n = [1, 3, 5, 7, 9, 11]$. The scales cover 3 1/2 octaves, although the scale differences are not uniform. Once the center-surround responses are computed at each position and scale, we find the extrema by comparing each point in the 3D image-scale space with its 26 neighbors in scale and position. With CenSurE features, a consensus match can be found for the outdoor images (Figure 2).

While the basic idea of CenSurE features is similar to that of SIFT, the implementation is extremely efficient, comparable to Harris or FAST detection [3][1]. We compared the matching ability of the different features over the 47K of the Little Bit dataset (see Section III). We tested this in two ways: the number of failed matches between successive frames, and the average length of a feature track (Table I second row). For VO, it is especially important to have low failure rates in matching successive images, and CenSurE failed on just 78 images out of the 47K image set (.17%). The majority of these images were when the cameras were facing the sky, and almost all of the image was uniform. We also compared the performance of these features on a short out-and-back trajectory of 150m (each direction) with good scene texture and slow motion, so there were no frame matching failures. Table II compares the loop closure error in meters (first row) and as percentage (second row) for different features. Again CenSurE gives the best performance in terms of the lowest loop closure error.

	Harris	FAST	SIFT	CenSurE
Fail	0.53%	2.3%	2.6%	0.17%
Length	3.0	3.1	3.4	3.8

TABLE I
MATCHING STATISTICS FOR THE LITTLE BIT DATASET

	Harris	FAST	SIFT	CenSurE
Err	4.65	12.75	14.77	2.92
%	1.55%	4.25%	4.92%	0.97%

TABLE II
LOOP CLOSURE ERROR FOR DIFFERENT FEATURES

D. Incremental Pose Estimation

The problem of estimating the most recent N frame poses and the tracked points can be posed as a nonlinear minimization problem. Measurement equations relate the points q_i and frame poses C_j to the projections q_{ij}, according to (1). They also describe IMU measurements of gravity normal and yaw angle changes:

$$g_j = h_g(C_j) \tag{5}$$

$$\Delta\psi_{j-1,j} = h_{\Delta\psi}(C_{j-1}, C_j) \tag{6}$$

The function $h_g(C)$ returns the deviation of the frame C in pitch and roll from gravity normal. $h_{\Delta\psi}(C_{j-1}, C_j)$ is just

[1]For 512x384 images: FAST 8ms, Harris 9ms, CenSurE 15ms, SIFT 138ms.

the yaw angle difference between the two frames. Given a set of measurements $(\hat{q}_{ij}, \hat{g}_j, \hat{\Delta\psi}_{j-1,j})$, the ML estimate of the variables C_j, q_i is given by minimizing the squared error sum

$$\min_{C_{j>n}, q_i} \left\{ \sum_{i,j} \epsilon_v^\top Q_v^{-1} \epsilon_v + \sum_{j-1,j} \epsilon_{\Delta\psi}^\top Q_{\Delta\psi}^{-1} \epsilon_{\Delta\psi} + \sum_j \epsilon_g^\top Q_g^{-1} \epsilon_g \right\} \tag{7}$$

Here the error terms are the differences between the measured and computed values for a given C_j, q_i. Note that we have only specified minimization over C_j for $j > n$, which holds the oldest $n < N$ frames constant to anchor the bundle adjustment. For the experiments of this paper, we used $N = 9$ and $n = 6$.

The IMU terms in (7) were introduced to show the optimal way to incorporate these measurements. In our current experiments, we instead use a simple post-filter that operates only on the latest frame (Section II-E). Considering just the first term, (7) can be linearized and solved by Taylor expansion, yielding the *normal equation*

$$J_q^\top(x) Q^{-1} J_q(x) dx = -J_q^\top(x) Q^{-1} q(x), \tag{8}$$

where x are the initial values for the parameters $C_{j>n}, q_i$, $q(x)$ are all the q_{ij} stacked onto a single vector, Q is the block-diagonal of Q_v^{-1}, and J is the jacobian of the measurements. After (8) is solved, a new $x' = x + dx$ is computed, and the process is iterated until convergence. Various schemes to guarantee convergence can be utilized, typically by adding a factor to the diagonal, as in the Levenberg-Marquadt algorithm.

The system (8) is very large, since the number of features is typically hundreds per frame. *Sparse bundle adjustment* efficiently solves it by taking advantage of the sparse structure of the jacobian. In our experiments (Section III), SBA can decrease the error in VO by a factor of 2 to 5 - this is the first large-scale experiment to quantify error reductions in VO from SBA (see [23], [16] for smaller experiments).

E. VO and IMU fusion using EKF

In this paper, we use VO to provide incremental pose estimation and the IMU as both an inclinometer (absolute roll and pitch) and an angular rate sensor (for incremental yaw). Translation estimates from VO are not corrected by IMU but will benefit from the improved angular precision (see discussion on drifts in Section II). We fuse VO and IMU via *loose coupling*, in which each positioning sub-system is taken as a pose estimator and treated as a black box. Pose information from each device is fused in a second stage with simple, small sized EKF procedures. Loose coupling is suboptimal in the sense that the existing cross-correlations between internal states of different devices are discarded. However, we show that long term VO accuracy can be dramatically improved by just using the simplest loosely coupled fusion with IMU information.

1) EKF formulation: The EKF formulation is quite straightforward. To ensure continuity and differentiability properties of the state vector, the vehicle's orientation is encoded with a quaternion, and the state of the vehicle is represented by the 7-vector pose $X = [x, y, z, a, b, c, d]^\top$. Its Gaussian character is specified by the couple $\{\hat{X}, P\}$ so that $p(X) = \mathcal{N}(X - \hat{X}, P)$, which is initialized to the null value $\{\hat{X}_0 = [0, 0, 0, 1, 0, 0, 0]^\top, P_0 = 0_{7\times 7}\}$. We call k the filter time index, which corresponds to the last SBA frame N. We systematically enforce quaternion normalization and Euler gimbal-lock corrections whenever necessary.

We start with EKF motion prediction from VO. At each iteration of the filter, we take the VO incremental motion Gaussian estimate $\{C_\Delta, Q_\Delta\}$, from frame $N-1$ to N, and use it for prediction via standard EKF.

Second we correct the absolute gravity normal by using the IMU as an inclinometer. Our IMU provides processed (not raw) information in the form of a vehicle pose $C_k^{IMU} = [x_k, y_k, z_k, \phi_k, \theta_k, \psi_k]^\top$ (position, and orientation in Euler angles). Accelerometers inside the IMU sense the accelerations together with the gravity vector. Because accelerations have zero mean in the long term, these readings provide absolute information about the gravity direction. The gravity vector readings in the vehicle frame only depend on *roll* ϕ_k and *pitch* θ_k angles, so we define the measurement equation (5) to be:

$$g_k = h_g(C_k^{IMU}) = \begin{bmatrix} \phi_k \\ \theta_k \end{bmatrix}. \tag{9}$$

Its uncertainty $G = diag(\sigma_g^2, \sigma_g^2)$ is given a large σ_g of around $0.5\mathrm{rad}$ to account for unknown accelerations. The Gaussian couple $\{g_k, G\}$ is then fed into the filter with a regular EKF update.

Finally we correct relative yaw increments by exploiting the angular rate readings of the IMU. The measurement equation (6) for the yaw increment in the IMU is trivial:

$$\Delta\psi_{k-1,k} = h_{\Delta\psi}(C_{k-1}^{IMU}, C_k^{IMU}) = \psi_k - \psi_{k-1} \tag{10}$$

This *yaw increment* is added to the last filter estimate $\hat{X}_{k-1}$ (that has been stored in the last iteration) to obtain a *yaw measurement* that is relative in nature yet absolute in the form, hence adequate for a classical EKF update:

$$y_k = h_\psi(\hat{X}_{k-1}) + \Delta\psi_{k-1,k} \tag{11}$$

where the observation function $h_\psi(X)$ provides the yaw angle of the pose X. Its noise variance is $Y_k = \sigma_{\Delta\psi}^2 \Delta t$, with $\sigma_{\Delta\psi}$ the angular random walk characteristic of the IMU (in $rad/\sqrt{s}$ units), and Δt the time lapse in seconds from $(k-1)$ to k. The Gaussian couple $\{y_k, Y_k\}$ is then fed into the filter with a regular EKF update.

III. EXPERIMENTS

We are fortunate in having large outdoor datasets with frame-registered ground truth from RTK GPS, which is accurate to several cm in XY and 10 cm in Z. For these datasets, the camera FOV is 35 deg, the baseline is 50 cm, and the frame rate is 10 Hz (512x384), so there is often

large image motion. We took datasets from Little Bit (9 km trajectory, 47K frames) in Pennsylvania, and Ft Carson (4 km, 20K frames) in Colorado, to get variety in imagery. The Ft Carson dataset is more difficult for matching, with larger motions and less textured images. In the experiments, we use only CenSurE features, which failed the fewest times (0.17% for Little Bit, 4.0% for Ft Carson).

A. VO Error Analysis and Calibration

We analyze VO-only drifts in order to provide useful odometry-like error statistics. The odometry model consists of translation and angular errors over displacement ($\mathrm{m}/\sqrt{\mathrm{m}}$ and $\mathrm{deg}/\sqrt{\mathrm{m}}$), and angular error over rotation ($\mathrm{deg}/\sqrt{\mathrm{deg}}$); but we do not compute the latter because of lack of data. From the 9km run of the Little Bit data, we select 100 sections 200m long, and integrate position and angular drifts. Averaging then reveals both random walk drifts and deterministic drifts or biases (Fig. 4 and summary in Table III). We observe a) a nearly ideal random walk behavior of the angular drifts; and b) a linear growth of the mean position drifts. This linear growth comes from systematic biases that can be identified with uncalibrated parameters of the visual system. X drift (longitudinal direction) indicates a scale bias that can be assigned to an imperfect calibration of the stereo baseline. Y and Z drifts (transversal directions) indicate pan and tilt deviations of the stereo head. The angles can be computed by taking the arc-tangent of the the drifts over the total distance. In Table IV we show the performance improvement of SBA – note the significant improvement in error values.

B. Trajectories

The VO angular errors contribute nonlinearly to trajectory error. On the two datasets, we compared RMS and max XYZ trajectory errors. In the case of matching failure, we substituted IMU data for the angles, and set the distance to the previous value. In Table V, the effects of bundle adjustment and IMU filtering are compared. Figure 5 has the resultant trajectories.

In both datasets, IMU filtering plays the largest role in bringing down error rates. This isn't surprising, since angular drift leads to large errors over distance. Even with a noisy IMU, global gravity normal will keep Z errors low. The extent of XY errors depends on how much the IMU yaw angle drifts over the trajectory - in our case, a navigation-grade IMU has 1 deg/hr of drift. Noisier IMU yaw data would lead to higher XY errors.

The secondary effect is from SBA. With or without IMU filtering, SBA can lower error rates by half or more, especially in the Ft. Carson dataset, where the matching is less certain.

IV. Conclusion

We have described a functioning stereo VO system for precise position estimation in outdoor natural terrain. Using a novel scale-space feature, sparse bundle adjustment, and filtering against IMU angle and gravity information, we obtain an extremely precise positioning system for rough terrain travel. The system is currently being integrated on a large autonomous outdoor vehicle, to operate as an alternative to GPS/IMU positioning. The error statistics of the VO/IMU described here are better than standard (non-RTK) GPS/IMU systems over ranges up to 10 km.

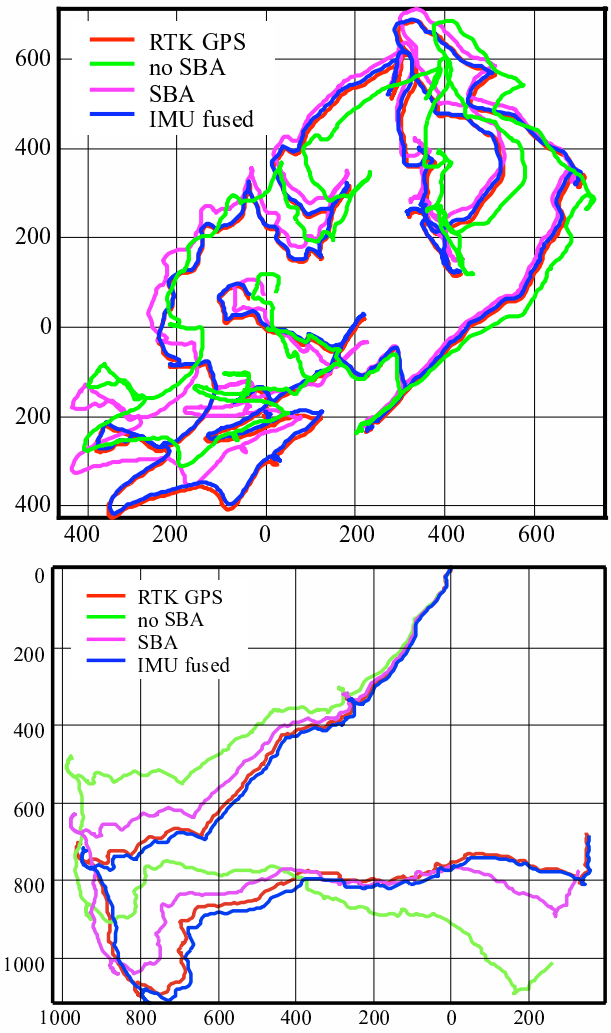

Fig. 5. Trajectories from Little Bit (top) and Ft Carson (bottom) datasets.

Currently the system runs at about 10 Hz on a 2 GHz Pentium, with the following timings: feature extraction (15 ms), dense stereo (25 ms), tracking (8 ms), RANSAC estimation (18 ms), SBA (30 ms), and IMU filtering (0 ms). We are porting to a dual-core system, and expect to run at 20 Hz.

One area of improvement would be to integrate a full model of the IMU, including forward accelerations, into the minimization (7). For datasets like Ft Carson, where there are significant problems with frame-to-frame matching, the extra information could help to lower the global error.

References

[1] M. Agrawal and K. Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive gps. In *ICPR*, August 2006.

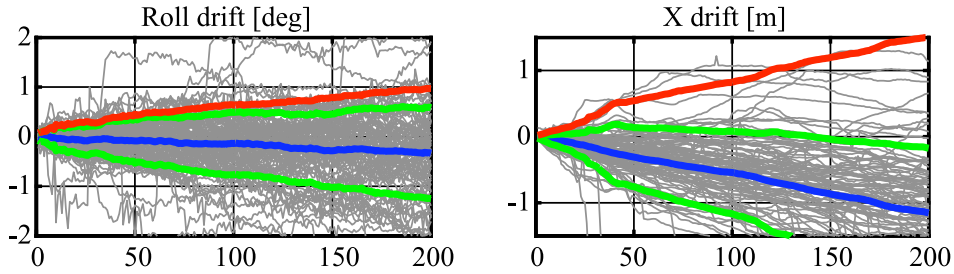

Fig. 4. Integrated errors for Roll (left) and X (right) estimates. All 200m long trajectories (grey), their mean (blue), STD (green, shown at $\pm 1\sigma$ from mean) and RMS (red). Couples {Pitch,Y} and {Yaw,Z} behave similarly.

state	mean	STD	derived calibration
	deg/m	deg/$\sqrt{\text{m}}$	
ϕ	-0.002	0.065	white noise: no info
θ	0.002	0.196	white noise: no info
ψ	-0.003	0.145	white noise: no info
	mm/m	mm/$\sqrt{\text{m}}$	
x	-5.76	69	-0.58% baseline factor
y	-11.1	59	0.63° stereo pan
z	11.9	34	0.68° stereo tilt

TABLE III
VO ERRORS (CENSURE, SBA, NO IMU)

state	no SBA	SBA
ϕ	0.191	0.065
θ	0.243	0.196
ψ	0.230	0.145
x	99	69
y	73	59
z	44	34

TABLE IV
STD VALUES FOR TWO VO METHODS (IN $\{\text{deg}, \text{mm}\}/\sqrt{\text{m}}$)

TABLE V
TRAJECTORY ERROR STATISTICS, IN METERS AND PERCENT OF TRAJECTORY

		RMS error in XYZ	Max error in XYZ
Little Bit	VO No SBA	97.41 (1.0%)	295.77 (3.2%)
	VO SBA	45.74 (0.49%)	137.76 (1.5%)
	VO No SBA + IMU	7.83 (0.08%)	13.89 (0.15%)
	VO SBA + IMU	4.09 (0.04%)	7.06 (0.08%)
Ft Carson	VO No SBA	263.70 (6.9%)	526.34 (13.8%)
	VO SBA	101.43 (2.7%)	176.99 (4.6%)
	VO No SBA + IMU	19.38 (0.50%)	28.72 (0.75%)
	VO SBA + IMU	13.90 (0.36%)	20.48 (0.54%)

[2] M. Agrawal and K. Konolige. Rough terrain visual odometry. In *Proc. International Conference on Advanced Robotics (ICAR)*, August 2007.

[3] M. Agrawal and K. Konolige. CenSurE: Center surround extremas for realtime feature detection and matching. In Preparation.

[4] M. Agrawal, K. Konolige, and R. Bolles. Localization and mapping for autonomous navigation in outdoor terrains: A stereo vision approach. In *WACV*, February 2007.

[5] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3), 1995.

[6] A. Comport, E. Malis, and P. Rives. Accurate quadrifocal tracking for robust 3d visual odometry. In *ICRA*, 2007.

[7] A. Davison. Real-time simultaneaous localisation and mapping with a single camera. In *ICCV*, pages 1403–1410, 2003.

[8] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE PAMI*, 29(6), 2007.

[9] C. Engels, H. Stewnius, and D. Nister. Bundle adjustment rules. *Photogrammetric Computer Vision*, September 2006.

[10] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. ACM.*, 24:381–395, 1981.

[11] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988.

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

[13] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *IEEE Conference on Image Processing (ICIP)*, 2002.

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

[15] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision (ECCV)*, 2002.

[16] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real time localization and 3d reconstruction. In *CVPR*, volume 1, pages 363 – 370, June 2006.

[17] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE PAMI*, 26(6):756–770, June 2004.

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

[19] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, volume 1, 2006.

[20] J. Shi and C. Tomasi. Good features to track. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 1994.

[21] J. Sola, M. Devy, A. Monin, and T. Lemaire. Undelayed initialization in bearing only slam. In *ICRA*, 2005.

[22] D. Strelow and S. Singh. Motion estimation from image and inertial measurements. *International Journal of Robotics Research*, 23(12), 2004.

[23] N. Sunderhauf, K. Konolige, S. Lacroix, and P. Protzel. Visual odometry using sparse bundle adjustment on an autonomous outdoor vehicle. In *Tagungsband Autonome Mobile Systeme*. Springer Verlag, 2005.

[24] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzibbon. Bundle adjustment - a modern synthesis. In *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.

[25] P. Viola and M. Jones. Robust real-time face detection. In *ICCV01*, 2001.