

# VideoTrek: A Vision System for a Tag-along Robot

Oleg Naroditsky    Zhiwei Zhu    Aveek Das    Supun Samarasekera    Taragay Oskiper  
Rakesh Kumar  
Sarnoff Corporation  
201 Washington Rd., Princeton, NJ 08540  
onaroditsky@sarnoff.com

## Abstract

*We present a system that combines multiple visual navigation techniques to achieve GPS-denied, non-line-of-sight SLAM capability for heterogeneous platforms. Our approach builds on several layers of vision algorithms, including sparse frame-to-frame structure from motion (visual odometry), a Kalman filter for fusion with inertial measurement unit (IMU) data and a distributed visual landmark matching capability with geometric consistency verification. We apply these techniques to implement a tag-along robot, where a human operator leads the way and a robot autonomously follows. We show results for a real-time implementation of such a system with real field constraints on CPU power and network resources.*

## 1. Introduction

In order to successfully perform real-world navigation tasks, autonomous mobile robots must be able to locate themselves in a previously explored environment. In the literature, collection of visual data from the environment is often done by the robot itself, either in exploration mode or during a training stage. The ability to use a platform with the same dynamic properties for exploration as for navigation is an unrealistic assumption for many tasks. Ground robotics vehicles come in a variety of sizes and holonomicity. If those robots are to cooperate either among themselves or with people, they need to be able to exchange visual information in a mutually understandable format. We propose a system that takes a step in that direction by allowing a mobile robot to follow a path automatically laid out by a human operator exploring the environment through the exchange of visual landmarks.

For global model based visual navigation, real-time, dead-reckoning visual odometry has been developed over the past few years [11, 10, 2]. Visual odometry systems seek to maintain the vehicle's 6-DOF pose in a global world



Figure 1. Our system in action during a leader-follower experiment. The robot, equipped with stereo cameras and an IMU, is autonomously following an operator wearing the same sensors on his helmet. The system uses visual navigation with robot-to-helmet landmark matching capability to achieve non-line-of-sight, tag-along robot capability. The operator can move freely in the environment, walking, running, looking around and walking backwards.

coordinate system (with respect to some initial known position). Some of the more recent approaches aim to combine the visual pose estimates with readings from IMU [13, 7], GPS [1] using Kalman filters. Pose estimates of such systems will eventually succumb to drift (unless GPS is used) or may experience errors due to problems with feature tracking. On the positive side, real-time implementations for a variety of camera configurations have been developed. Recently, real-time schemes that include sparse bundle adjustment have been proposed by [3, 8].

To avoid the effects of drift of dead reckoning systems, techniques in topological SLAM, visual servoing and global place recognition are used. Popular appearance-based ap-

proaches that seek to do global loop closing and recognition rely on SIFT features [9] quantized into vocabulary trees proposed by [12]. Examples of such systems, which also incorporate geometric consistency, are given in [4, 16]. Another approach is to directly match features between images using wide baseline algorithms and directly recover the vehicle’s position with respect to target [5]. A method for navigation using local feature graphs and visual servoing is proposed in [14].

All of the systems where a mobile robot was required to re-traverse a path have been developed with the image data collected by the same platform either during an exploration stage or during human-controlled training stage. On the other hand, we describe a system where two totally different platforms can do SLAM in the same environment.

We develop a multilayer navigation system which we call VideoTrek (shown in action in Figure 1) which incorporates elements of model-based and appearance-based systems. First, we compute a highly accurate, distributed aperture visual odometry pose solution. This pose is then augmented with readings from the IMU. We then use a vocabulary tree of quantized histogram of oriented gradients (HOG) features (landmarks) to maintain location fingerprints. While the global drift introduced by dead reckoning algorithms is an important factor in applications, such as place recognition and loop closing, it is not a big factor in ours. A tag-along robot must only maintain its pose with respect to the leader’s traversed path, not the global coordinate system. Thus the visual odometry pose only serves as input to global landmark matching and to maintain the vehicle’s pose in the short term in absence of landmark matches. The novel feature of our application is in the live, automatic sharing of visual landmarks between the operator wearing a sensor-equipped helmet and an autonomous robot. We also investigate the system’s performance through a series of real world experiments that highlight its accuracy and robustness.

## 2. Vision system components

The vision systems on the leader (helmet) and follower (robot) consist of 4 wide field of view cameras, arranged in 2 stereo pairs with one looking forward and the other backward. Each system also has an inexpensive IMU sensor which provides local orientation rates at 100Hz. Figure 2 contains a diagram of our system.

### 2.1. Dead reckoning visual navigation

The foundation of our navigation system is robust structure from motion estimation using a modified stereo scheme from [11]. Instead of a frame-to-frame pose stitching, we employ a dynamic local landmark matching scheme from [15] where feature tracks are maintained to a chosen refer-

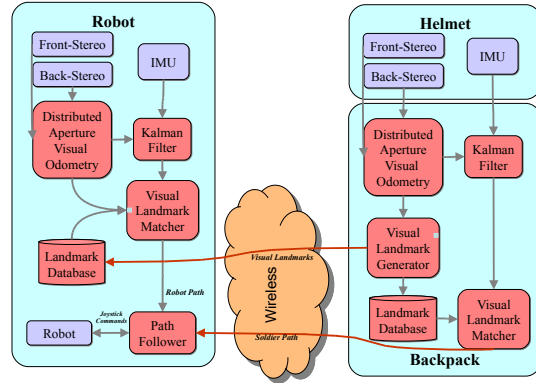


Figure 2. A chart showing major hardware and software components the VideoTrek system. Hardware devices are blue, and algorithm blocks are red.

ence frame for as long as possible (until an unfavorable 3D distribution of features is detected) to minimize drift accumulation during small motions of the platform. These pose estimates are computed at a rate of 15Hz on our system and converted into frame-to-frame estimates. Since each navigation system uses two mutually-calibrated stereo pairs, we use the distributed aperture technique first described by Oskiper, et al [13].

For a stereo frame  $k$ , we extract Harris corner locations from the left and right images. The patches around the feature locations are then matched, the matches transformed into the camera coordinate system and triangulated to produce a set of 3D points in the left camera’s coordinate system. Some of the errors are immediately corrected with epipolar geometry and left-right checks based on the extrinsic calibration. For the next pair of images  $k + 1$ , we establish temporal correspondences of the Harris corner locations with the frame  $k$ . We now have a collection of 2D-to-3D correspondences, from which we can compute the pose of the camera at  $k + 1$  by using a RANSAC process with hypotheses being generated with a 3-point 3D resection algorithm. Hypothesis scoring and the refinement is done based on the reprojection error of the points in both left and right images with a robust Cauchy cost function also described in [13]. The winning hypothesis is then refined with an iterative refinement process. After the pose is established, the process repeats for frame  $k + 1$ , starting with re-triangulation.

After each stereo pair is processed, we robustify the visual odometry process even further by selecting the best estimate of the two available global scores.

Even with multiple cameras, there are still situations (al-

though greatly minimized) where visual odometry alone provides poor pose estimates. Therefore, in order to further increase the robustness of our system we integrated our system with a MEMS based IMU using the filter model suggested by [13], which we will briefly summarize here. The state vector for this constant velocity filter contains 16 elements:  $X$ , (3-vector) representing position in navigation coordinates,  $q$ , unit quaternion (4-vector) for attitude representation in navigation coordinates,  $v$ , (3-vector) for translational velocity in body coordinates,  $\omega$ , (3-vector) for rotational velocity in body coordinates, and,  $b$ , (3-vector) for angular rate sensor component (gyro) biases of the IMU. We now define our process model as

$$X_k = X_{k-1} + R^T(q_{k-1})x_{\text{rel}}, \quad (1)$$

$$q_k = q_{k-1} \otimes q(\rho_{\text{rel}}), \quad (2)$$

$$\omega_k = \omega_{k-1} + n_{\omega,k-1} \quad (3)$$

$$b_k = b_{k-1} + n_{b,k-1} \quad (4)$$

$$v_k = v_{k-1} + n_{v,k-1} \quad (5)$$

where

$$x_{\text{rel}} = v_{k-1}\Delta t_k + n_{v,k-1}\Delta t_k \quad (6)$$

$$\rho_{\text{rel}} = \omega_{k-1}\Delta t_k + n_{\omega,k-1}\Delta t_k \quad (7)$$

and  $\otimes$  is the quaternion product operation. The rotation vector  $\rho$  is in the body frame,  $R(q)$  is the rotation matrix determined by the attitude quaternion  $q$  in the navigation frame, and is the quaternion  $q$  obtained from the rotation vector. Undetermined accelerations in both translational and angular velocity components and the bias process noise are modeled by zero mean white Gaussian noise processes  $n$ . The filter runs at the frame rate, meaning that the discrete time index denoted by  $k$  corresponds to the frame times for which pose outputs are available from visual odometry.

The gyro and accelerometer readings from the IMU are used as measurements in the Kalman filter. Integrating all the intermediate gyro velocities between consecutive video frame time instants, rotational velocities are derived for the frame to frame rotational motion. The multi-camera visual odometry frame to frame local pose measurements expressed in the coordinate frame of the front left camera,  $P_k = P(t_k, t_{k+1})$ , are also converted to velocities by extracting the rotation axis vector corresponding to the rotation matrix  $R_k$ , together with the camera translation given by  $R^T T_k$ , (where  $P_k = [R_k | T_k]$ ) and then dividing by the timestep,  $\Delta t_k = t_{k+1} - t_k$ . The accelerometer data corresponding to frame instants is obtained by interpolating the two accelerometer readings that arrive right before and after every frame and this information is used only when the body acceleration is below a certain threshold to avoid contamination. Hence, the observations from visual odometry

and IMU are used according to the following measurement model:

$$v_k^{\text{vo}} = v_k + n_{v,k}^{\text{vo}}, \quad (8)$$

$$\omega_k^{\text{vo}} = \omega_k + n_{\omega,k}^{\text{vo}}, \quad (9)$$

$$\omega_k^{\text{imu}} = \omega_k + b_k + n_{\omega,k}^{\text{imu}}, \quad (10)$$

$$a_k^{\text{imu}} = R(q_k)g + n_{a,k}^{\text{imu}}. \quad (11)$$

Here,  $v^{\text{vo}}$  and  $\omega^{\text{vo}}$  are translational and angular velocity measurements provided by visual odometry (vo), and  $\omega^{\text{imu}}$ , and  $a^{\text{imu}}$  are the gyro and accelerometers outputs provided by the IMU, and  $g$  is the gravity vector. Uncertainty in the visual odometry pose estimates, represented by the noise components is estimated based by the reprojection error covariance of image features through backward propagation. 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 and also to estimate the gyro component biases using the good measurements from visual odometry that come with high confidence.

## 2.2. Distributed landmark-based visual navigation

The high rate, fast dead reckoning pose is sufficient for the robot controller to execute maneuvers in a local coordinate system. In fact, the controller operates on the 3 degree of freedom velocity estimate extracted from visual odometry. If we want the robot to follow a human, the path planning stage requires an estimate of robot's pose in the human's coordinate system. This is accomplished through the use of visual landmarks. Both the leader and the follower proceed by the method proposed by Zhu et al. [16] where the sparse 3D interest points (triangulated from Harris corner locations) from the visual odometry process are used to create HOG descriptors. The scale for these descriptors is fixed to be proportional to the depth of the interest point viewed from the left image of each stereo pair. The set of descriptors, along with the 3D positions in the world of the corresponding points and time stamp of the original image is bundled into a data stream we call a "landmark snapshot". These snapshots serve as location fingerprints since they encode both the visual and geometric information about the scene, and are therefore quite unique.

At this point the leader and the follower's tasks diverge. It is only necessary for the leader to maintain its relative pose (relative to the start of the path), but the follower must maintain its pose in relation to the leader. To this end, the

leader sends its landmark snapshots to the follower. The follower then uses the leader's snapshots to build up a map of the environment by quantizing the feature descriptors into a vocabulary tree [12] structure in follower's memory, with the descriptors being accessible via the inverted file structure. The features' 3D positions and the timestamp are also stored in the database. There are around 100 to 200 features per frame that pass the visual odometry's inlier criteria and are inserted into the database.

The follower's snapshots are then matched to the database built with the leader's features. From the top matches we then select only ones taken from a location with the Euclidean distance closer than 5m to the follower's present location. We then estimate the geometric consistency for the remaining top matches in reverse order of insertion into the database (based on the stored time stamp). The ordering is there simply because we prefer to match to the most recent location sighting, up to some point in the past. The consistency check is done by estimating the relative pose of the follower's camera with the 3D features collected by the leader. The set of inlier features is then computed based on the reprojection error. If the number of inliers is greater than a threshold, the match is considered to be successful and the newly-computed final pose  $P_{\text{final}}$  is output. Since dead reckoning and landmark matching operate in parallel on our system, we must compute the pose correction as  $P_c = P_f^{-1}P_{\text{final}}$  where  $P_f$  is the follower's pose at the time that the matched frame was captured. This correction, which is, essentially, an accumulated error in pose between the leader and follower, is then applied as an offset to all subsequent poses until a new match is found and a new correction calculated.

### 3. Motion planning and control

The corrected globally aligned 6 DOF pose measurements from both the helmet and the robot visual processing blocks are projected into ground aligned 3 DOF poses (planar position and orientation) for robot motion planning and control. Since the coordinate systems are aligned, we use the leader's trajectory to generate the path plan for the follower after appropriate (for robot path tracking) processing for smoothness and continuity. The follower (robot) uses a nonlinear steering controller [6] for path following using cross track error as feedback. The controller also compensates for delays in the low level actuators and slows down the robot on tight curves to reduce slip. The path planning and vehicle control run asynchronously with landmark-corrected visual pose estimation. The maximum speed of the follower robot was set at 0.8m/s with lower speeds during turns.



Figure 3. The VideoTrek follower system installed on a mobile robot. The sensors are in the white boxes on the platform and the processor is in a black box visible on the side of the robot.



Figure 4. The VideoTrek leader system. The cameras are visible on the sides of the helmet. The gray cable connects the sensors on the helmet with the processing inside the backpack.

### 4. System integration

In the VideoTrek system, the leader, shown in Figure 4 and the follower, shown in Figure 3, use identical sensors. Each system has two stereo pairs, looking forward and backward (with respect to the dominant motion direc-

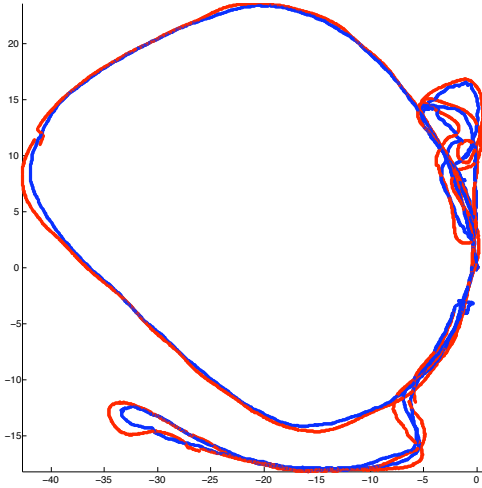


Figure 5. A plot of leader's trajectory (blue) for the "Circle 2" experiment and follower's estimated position with respect to the leader's trajectory (red) based on follower's dead reckoning and visual landmark matching.

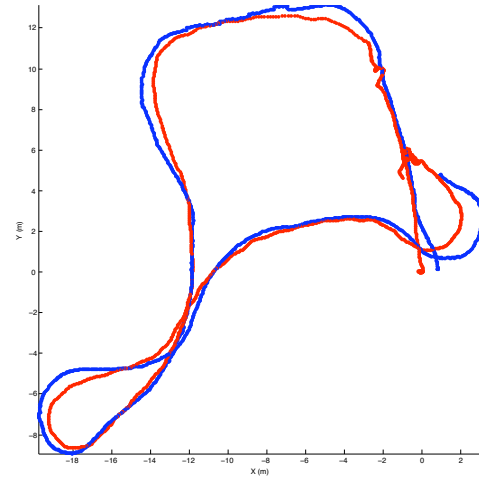


Figure 6. Trajectory for the leader (blue) and follower (red) for one of the Repeatability Experiment runs from Table 2.

tion of the platform), and an inexpensive IMU unit. The stereo baseline and inter-stereo pose are different for each system due to different packaging requirements of the helmet and robot systems. The baseline is 17cm for the robot's stereo cameras, and 23cm for the helmet. The 640x480 pixel FireWire cameras are used with 70 degree field of view lenses. The leader's sensors are built into a helmet with the PC residing inside a backpack and the follower system is integrated with a mobile robot. In both cases the cameras were pointing toward the ground at an angle of approximately 15 degrees from horizontal in order to capture nearby features.

Each system contains an Intel CoreDuo-based PC, but image processing rates are different on the leader and the follower, according to the tasks. The helmet-based leader system needs to have high frame rate to keep up with fast head motions, and thus was operating dead reckoning navigation at 15Hz, while the slower moving robot system operated at 10Hz, and the extra processing time was used by path planning and robot control. Both systems generate and matched landmarks asynchronously at 1Hz.

Each system is equipped with an 802.11-based wireless network capability used for path and landmark communication. The average bandwidth use with our system is about 100KB/s, which corresponds to the size of an average landmark snapshot.

#### 4.1. System operation

The VideoTrek system was field-tested with the following procedure for each experiment:

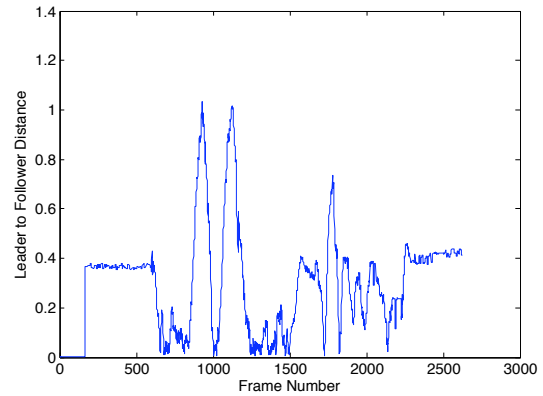


Figure 7. Estimated distance between the leader and follower, based on landmark matching between the two for one of the repeatability experiments from Table 2. This demonstrates that the drift is kept in check by the matching. The initial 167 frames occurred before the coordinate systems synchronized.

1. The operator puts on the helmet system and stands next to the robot.
2. The operator activates the navigation systems of the leader and the follower via a tablet computer interface.
3. The operator moves his head until the field of view of the helmet and the robot overlap sufficiently to establish a landmark match, synchronizing the coordinate systems. The operator is notified of this event.
4. The operator is now free to move about in the environment, making sure to move only in places where the

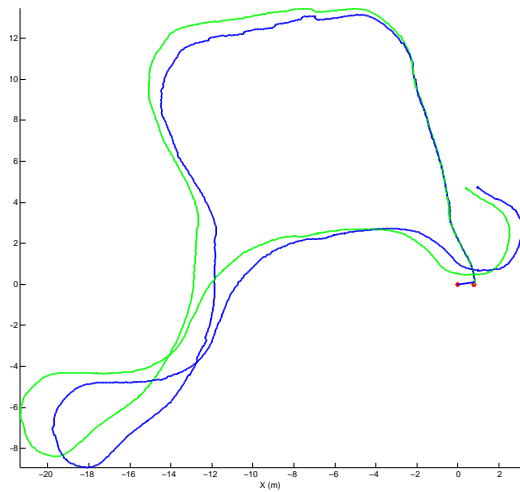


Figure 8. Trajectory shape comparison between the landmark-corrected robot path (blue) and Kalman filter output (green). Without landmark correction, the robot’s visual navigation calculated it was taking the green path. (Note: the green trajectory is not necessarily the path the robot would have taken if landmark matching was not present.)

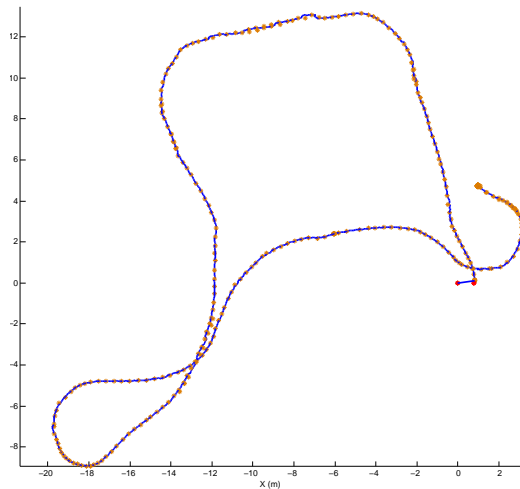


Figure 9. Landmark-corrected robot path (blue) with landmark match locations indicated by stars. The match locations are not necessarily on the path because the corrections take several frame times to calculate and consequently can only be applied to several frame times in the future.

robot can operate safely. He can look around freely, kneel, run, walk backwards and sideways.

5. Once the operator comes to a stop, the robot traverses the remaining distance, and then stops at a safe distance from the operator. If the operator chooses to start moving again, the robot will follow.
6. Upon reaching the destination, the operator deactivates the system via the tablet computer interface.

At the beginning of each experiment, the operator wearing the helmet should stand alongside the robot for the initial synchronization to take place. This makes it easy to overlap the fields of view of the robot and helmet. The synchronization usually happens within a few seconds following activation.

## 4.2. Results

The system was evaluated with respect to follower accuracy and robustness to leader’s pose change. We did not evaluate the leader’s absolute pose accuracy since long-term drift in leader’s pose does not result in degradation of the follower’s performance. The goal was to allow the leader the maximum freedom of motion, so in many of the experiments the operator looked around the environment freely, turned around, stopped, walked backwards and sideways, jogged and crouched. A qualitative assessment shows that the system recovers well from these disturbances. The visual navigation performance was evaluated by computing the distance between the leader’s position and follower’s position in the leader’s coordinate system. This distance relies on the follower to correctly estimate its pose with respect to the leader. Dead reckoning drift due to lack of matches as well as mismatches will result in large relative distances since the robot’s controller tries to maintain a minimum possible distance between leader and follower (with a safety distance around the operator where the robot cannot go). Two experiments with duration of about 11 minutes and 15 minutes respectively are shown in Figures 5 and 6. The plots shows the leader’s position in red and follower’s position in blue. The discontinuities in the robot’s position result from landmark matches, which adjust the robot’s notion of where it is. We cannot expect the robot’s controller to follow the operator’s path exactly and maintain speed during turns, which accounts for some amount of deviation from the path. The landmark match locations are shown with stars in Figure 9. Note that due to time taken to compute the matches, the pose corrections are not applied for several frames after an image is taken.

The system underwent extensive field testing for performance as well as reliability. It proved to be quite reliable throughout the day and on multiple surfaces, including sand, asphalt and grass. Another major issue for the system

Trial Name	Duration (s)	Length (m)	Avg. Error (m)	Max. Error (m)	Avg. Speed (m/s)
Long 1	1101	868	0.21	1.9	0.53
Long 2	1187	673	0.25	2.5	0.54
Loops	942	400	0.15	0.81	0.64
Circle 1	925	346	0.15	1.3	0.64
Circle 2	641	309	0.22	1.2	0.70
Retrace 1	700	305	0.18	1.3	0.66
Retrace 2	725	280	0.11	0.89	0.56
Retrace 3	454	135	0.22	0.96	0.62
Desert 1	658	229	0.34	2.1	0.50
Desert 2	327	106	0.31	1.1	0.60

Table 1. Experimental evaluation statistics. The duration column is the amount of time the robot took to traverse the path. The time duration includes any operator pauses during which navigation was running. The path length column is the distance travelled by the robot. Total distance travelled by the operator was a few meters greater since the safety system stopped the robot when it achieved a distance of 2m from the operator. Average speed refers to the forward speed of the robot during following.

Date/Time	Duration (s)	Length (m)	Avg. Error (m)	Max. Error (m)	Avg. Speed (m/s)
2008.09.21/16.37.16	285	87	0.25	0.88	0.51
2008.09.21/16.42.48	260	82	0.26	0.97	0.51
2008.09.21/17.14.02	293	85	0.29	1.0	0.51
2008.09.21/21.19.13	409	91	0.17	0.68	0.51
2008.09.22/18.53.31	331	83	0.32	0.93	0.51
2008.09.22/19.00.26	274	79	0.24	1.1	0.52
2008.09.22/21.34.27	397	81	0.14	0.95	0.50
2008.09.23/17.09.59	267	84	0.28	0.97	0.52
2008.09.23/17.40.56	261	85	0.29	1.22	0.51
2008.09.23/19.34.46	283	87	0.26	1.15	0.52
2008.09.23/20.20.51	263	84	0.28	0.85	0.51
2008.09.23/20.49.22	262	82	0.27	0.94	0.51
2008.09.24/14.20.35	266	85	0.28	0.97	0.52
2008.09.24/16.33.17	255	79	0.20	0.75	0.43
2008.09.24/17.29.18	271	86	0.27	0.92	0.52
2008.09.24/20.15.03	275	90	0.26	0.88	0.50
2008.09.24/21.02.18	264	91	0.27	0.91	0.51
2008.09.25/17.16.14	272	88	0.24	0.90	0.51
2008.09.25/18.07.13	317	87	0.27	0.85	0.50
2008.09.25/20.06.08	373	94	0.19	1.03	0.52
2008.09.25/21.01.20	243	88	0.30	1.10	0.51

Table 2. Repeatability experiments consisted of the system traversing the same route over 5 days at different times during the day.

is its handling of different height disparities between the operator and the robot. The helmet was worn by people with heights ranging from 1.62m to 1.83m, giving us a disparity range of 0.42m to 0.63m without impacting performance. The most common cause of failure during field demonstrations was loss of power, followed by the loss of network connectivity. The robot never visibly strayed from the path traversed by the operator.

Time synchronization between sensors within each multi-sensor rig is of great importance. By employing external triggering for the cameras and the IMU, the images

and IMU readings were synchronized within a few milliseconds. The tightly-coupled Kalman filter does not perform well if the input data is more than 5ms out of synchronization.

In practice the landmark matching system never confused one place for another, even on asphalt. This can be explained by the abundance of Harris corners (even in the aforementioned environments) combined with the discriminative power of HOG features and uniqueness of 3D configurations of features (even planar). In areas with a shortage of usable landmarks (such as when a person is looking off

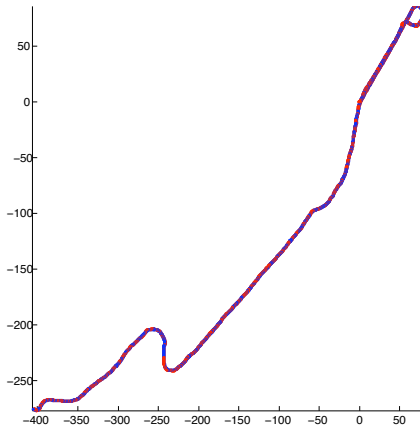


Figure 10. Path of the 868m leader (blue) and follower (red) run.

to the side), the dead-reckoning visual odometry system allows the robot to follow for 10s of meters (see [13] for a quantitative evaluation of this system) without registering a landmark match. See Figure 8 for an example of Kalman filter only path.

Figure 7 shows the distances (in the main motion plane) between the robot's and operator's perceived positions.

The Table 2 summarized results from 21 nearly identical runs in a desert environment, which constituted an official evaluation of the prototype. These runs took place during various times of the day. Other runs, with several different operators and including ones in suburban environment (see Figure 1) are shown in Figure 1. The longest recorded run of 868m is shown in 10. These results clearly show robustness and consistent performance under a variety of circumstances.

## 5. Conclusions

We presented algorithms and a system for a basic autonomous tag-along robot, capable of following the visual landmark trail sent to it automatically and in real time by an operator. While this system is in the prototype stage, it demonstrates the potential for relieving the burden on a human operator of a future robot in a very natural way. Equipped with obstacle avoidance, such a robot can follow any moving platform, forming a convoy of different robots and people. This robot also knows enough about the environment to retrace its steps, which is a useful capability (and a subject of research) that is naturally accomplished within our framework.

## References

[1] M. Agrawal and K. Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive gps. *International Conference on Pattern Recognition (ICPR)*, Jan 2006.

[2] M. Bansal, A. Das, G. Kreutzer, J. Eledath, R. Kumar, and H. Sawhney. Vision-based perception for autonomous urban navigation. *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pages 434–440, 2008.

[3] C. Engels, H. Stewenius, and D. Nister. Bundle adjustment rules. *Photogrammetric Computer Vision (PCV)(September 2006)*, 2006.

[4] F. Fraundorfer, C. Engels, and D. Nister. Topological mapping, localization and navigation using image collections. *Intelligent Robots and Systems*, Jan 2007.

[5] T. Goedeme, T. Tuytelaars, L. V. Gool, G. Vanacker, and M. Nuttin. Feature based omnidirectional sparse visual path following. *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1806 – 1811, Jul 2005.

[6] G. Hoffmann, C. Tomlin, M. Montemerlo, and S. Thrun. Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. *American Control Conference, 2007. ACC '07*, pages 2296 – 2301, Jun 2007.

[7] A. Howard. Real-time stereo visual odometry for autonomous ground vehicles. *Intelligent Robots and Systems*, Jan 2008.

[8] K. Konolige and M. Agrawal. Frameslam: From bundle adjustment to real-time visual mapping. *Robotics, IEEE Transactions on*, 24(5):1066 – 1077, Oct 2008.

[9] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, Jan 2004.

[10] M. Maimone, Y. Cheng, and L. Matthies. Two years of visual odometry on the mars exploration rovers. *JOURNAL OF FIELD ROBOTICS*, Jan 2007.

[11] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 1:I-652 – I-659 Vol.1, Jan 2004.

[12] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. *CVPR*, pages 2161–2168, 2006.

[13] T. Oskiper, Z. Zhu, S. Samarasekera, and R. Kumar. Visual odometry system using multiple stereo cameras and inertial measurement unit. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1 – 8, May 2007.

[14] S. Segvic, A. Remazeilles, A. Diosi, and F. Chaumette. Large scale vision-based navigation without an accurate global reconstruction. *Computer Vision and Pattern Recognition*, Jan 2007.

[15] Z. Zhu, T. Oskiper, O. Naroditsky, and S. Samarasekera. An improved stereo-based visual odometry system. *Proceedings of the Performance Metrics for Intelligent Systems (PerMIS'06)*, Jan 2006.

[16] Z. Zhu, T. Oskiper, S. Samarasekera, R. Kumar, and H. Sawhney. Real-time global localization with a pre-built visual landmark database. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 – 8, May 2008.