2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Robust Visual Path Following for Heterogeneous Mobile Platforms

Aveek Das* Oleg Naroditsky Zhiwei Zhu Supun Samarasekera Rakesh Kumar

Abstract— **We present an innovative path following system based upon multi-camera visual odometry and visual landmark matching. This technology enables reliable mobile robot navigation in real world scenarios including GPS-denied environments both indoors and outdoors. We recover paths in full 3D, making it applicable to both on and off-road ground vehicles. Our controller relies on pose updates from visual odometry, allowing us to achieve path following even when only a joystick drive interface to the base robot platform is available. We experimentally investigate two specific applications of our technology to autonomous navigation on ground vehicles - non line-of-sight leader-following (between heterogeneous platforms) and retro-traverse to home base. For safety and reliability we add dynamic short range obstacle detection and reactive avoidance capabilities to our controller. We show the results for end-to-end real time implementation of this technology using current off-the-shelf computing and network resources in challenging environments.**

I. INTRODUCTION

Recent advances in computer vision based algorithms and the increased availability of low-cost sensors and computers have accelerated the development of useful autonomous navigation behaviors for field deployed mobile robots. A key capability needed by autonomous mobile robots to perform real-world navigation tasks is to locate themselves in a previously explored environment. While the current state-of-art requires collection of visual data from the environment during exploration by the robot itself, in practice, we may need to have heterogeneous platforms (or even human agents carrying sensors) perform the exploration and subsequent path following tasks. If these robots are to cooperate either among themselves or with people, they need to be able to exchange visual information in a mutually understandable format. Toward this goal, we propose a system that extracts 3D visual landmarks and exchanges them using a unique visual database structure, allowing automatic alignment of paths explored by a leader to reference paths for a follower robot. We investigate two application scenarios: (1) *Retro-traverse* - the robot retraces its own path back to a designated home base location, and (2) *Leader-follower* - the robot follows the path traced by a human operator wearing a sensor pack without requiring direct line-of-sight.

Over the past few years, real-time visual odometry has been developed as an efficient dead-reckoning system for global model based visual navigation using a variety of camera configurations [1], [2], [3]. Visual odometry systems seek to maintain the vehicle's 6 DOF pose in a global world coordinate system (with respect to some initial known position). In order to deal with occlusions and failures in feature tracking, some of the more recent approaches combine the visual pose estimates with readings from inertial measurement units (IMU) [4], [5] and GPS [6] using Kalman filters. Pose estimates of such systems will eventually succumb to drift in GPS denied environments. Recently, real-time schemes that include sparse bundle adjustment simultaneously over a longer sequence of image frames of have been proposed by [7], [8], [9]. While these lower the drift rate, they introduce some latency in the final pose output.

Sarnoff Corporation, 201 Washington Rd., Princeton, NJ 08540
*adas@sarnoff.com

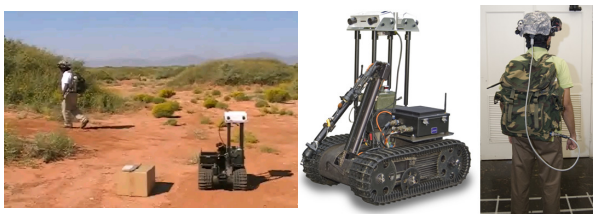

Fig. 1. (Left) The ViewTrek system in action during a leader-follower experiment. (Center) Mobile robot follower - The sensors are in the white boxes on the platform and the processor is in a black box visible on the side of the robot. (Right) Helmet leader - The cameras are visible on the sides of the helmet. The gray cable connects the sensors on the helmet to the processor inside the backpack.

To avoid the drift effects of dead-reckoning systems, techniques in topological SLAM, visual servoing and global place recognition have been used. Popular appearance-based approaches for global loop closing and recognition rely on SIFT features [10] quantized into vocabulary trees proposed by [11]. Examples of such systems, which also incorporate geometric consistency, are given in [12], [13], [14]. Another approach is to directly match features between images using wide baseline algorithms and directly recover the vehicle's position with respect to a target or home image [15]. A method for navigation using local feature graphs and visual servoing is proposed in [16] with an application to outdoor path following as shown in [17].

While some of the current visual path following techniques closely couple the SLAM module with the actual controller, others rely on just visual cues for localization and mapping, allowing well tested nonlinear controllers to achieve robust path tracking performance. Recently, the DARPA Grand Challenge provided a good evaluation test bed for the performance of path following controllers [18] for off-road driving at low and moderate speeds. We adapt the asymptotically stable nonlinear kinematic steering controller from [19] for our specific vehicle model and system dynamics.

In all related work where a mobile robot was required to re-traverse a path, path following strategies have been developed utilizing image data collected by the same platform during a prior exploration stage. Our system has the *unique* additional capability that two different mobile platforms

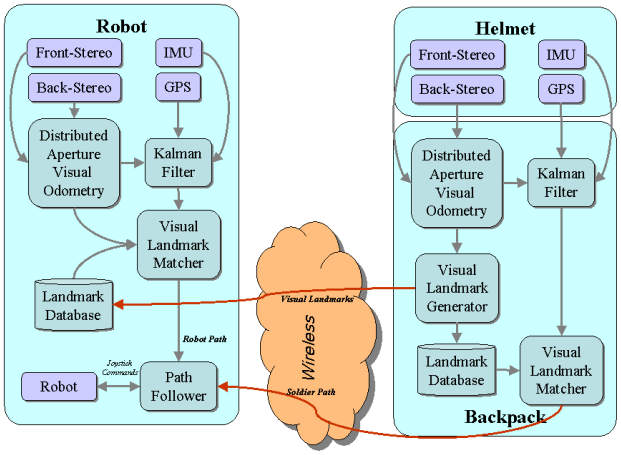

Fig. 2. A chart showing major hardware and software components of the *ViewTrek* leader-follower system. Hardware devices are purple, and algorithm blocks are blue. For retro-traverse the robot can be thought of as its own leader with no wireless networking requirement, but needing an additional software module for path reversal.

can do SLAM in the same environment. Further, the only interface between our visual navigation system and our path following control is the corrected pose output from the visual navigation system. We use the term visual path following since we use sparse visual features both for frame-to-frame motion estimation (visual odometry) as well as for global path alignment with drift correction (landmark matching).

We call our multilayer visual navigation system *ViewTrek* (shown in action in Figure 1) which incorporates elements of model-based and appearance-based systems. In the following sections, we first give a brief description of the visual navigation system (see [20] for vision algorithm details) followed by our motion control and reactive planning strategies. We then provide some system integration and operational details, followed by evaluation of the system's performance through a series of real world experiments.

II. Visual Navigation 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. Figures 2 show the functional diagram of our system for the leader-follower application.

A. Visual Odometry

Our visual odometry system consists of a feature-based, structure-from-motion scheme for distributed aperture stereo rigs [4]. For each image frame captured, this system produces a single 6 DOF projective transformation P, which defines the position and orientation (pose) of a coordinate frame the front left camera in the rig with respect to the previous image frame. At a given time instant t_{k+1}, we have $P_k = P(t_k, t_{k+1})$. This system operates as follows: for each calibrated stereo frame Harris corners are computed, and 3D points are triangulated. The pose of the current frame is computed by a robust resection process, using the Harris corner locations of the current frame and the corresponding 3D features computed from the previous frame. From the two estimates (one for each stereo pair), the best one is selected based on its error with respect to all features in the system [4].

The visual odomery process is enhanced by the readings from the IMU via a Kalman filter framework described in [4]. This filter uses the estimated covariances of the visual and inertial measurements to improve the overall pose estimate. The output of this filter is a pose estimate with respect to the location of the first image frame. We refer to these readings as dead-reckoning visual odometry. The multi-camera visual odometry frame to frame local pose measurements P_k 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, $\Delta t_k = t_{k+1} - t_k$. These velocity measurements enable kinematic path following control without requiring low-level interfaces with the robot actuators.

B. Distributed Landmark Matching

Once the robot is equipped with the camera rig, the camera pose output from visual odometry allows the robot controller to execute maneuvers on the ground in the camera coordinates system. However, this is a *local* coordinates system and if we want the robot to follow a human who wears a helmet-based system, 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. As in [13], the visual landmarks are selected from a set of interesting points extracted from the scene and each landmark is described by a Histogram of Gradients (HOG) descriptor in the image. These visual landmarks serve as the location fingerprints since they encode both the visual and geometric information about the scene, and are therefore quite unique. For simplicity, we refer to the robot's system as the follower and the human's system as the leader.

During the operation, in order to maintain the follower's pose in relation to the leader, the leader sends its extracted visual landmarks to the follower. The follower then uses the received visual landmarks to build up a landmark database, which will serve as a (sparse) map of the leader's environment. Specifically, the landmark database is indexed via a vocabulary tree structure [13] in the follower's memory for fast landmark matching. Given an image snapshot from the follower, it will be searched against the landmark database to find a match. When a successful landmark match happens, a pose P_{final} that aligns the follower into the leader's coordinates system will be generated. Since the visual odometry and landmark matching operate in parallel on our system, we must compute a 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 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. For retro-traversal the robot

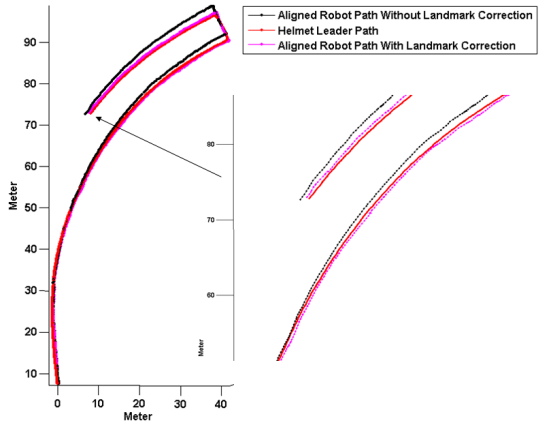

Fig. 3. Drift correction of visual odometry by matching against previously collected visual landmark database. The helmet operator walks along a *marked* path and collects the landmark database. The robot then loads in this database and runs visual navigation while being manually driven along the same path.

acts as its own virtual leader and locally stores the landmark database during the forward journey an matches against it on the way back. The forward-backward camera configuration was chosen as it increases the probability of finding landmark matches during path following. The effect of drift correction using landmark matching is shown for a controlled path in Figure 3.

III. Robot motion control and reactive planning

The landmark corrected globally aligned 6 DOF pose measurements P_{final} from both the helmet and the robot visual processing blocks are projected into ground aligned 3 DOF poses (planar position (x, y) and orientation θ) for robot motion planning and control. For leader following, since the coordinate systems are aligned, we use the leader's trajectory to generate the path plan for the follower. For retro traverse, the path is always aligned with respect to the specified start position.

A. Nonholonomic path following controller

Since most real mobile robots are nonholonomic we choose a representation for the paths that is C^2 smooth - natural cubic splines. We choose a discretization (usually $0.5m$) for the collected pose points and fit splines through them, represented by knot points. This representation allows us to efficiently locate points on the path as well as evaluate local tangents and normals. While generating the splines we also do some minimal spike suppression to reject sudden jumps in pose - this may happen when a there is a landmark based pose correction after a long interval, or when the helmet system makes very small looping motions.

As in [19], due to the low speeds of operation we use a kinematic vehicle model. We choose to use a car-like model even though our mobile robots may be tracked or skid-steer platforms, since this is the limiting case (based on turning radius and maximum steering angle constraints). Further, even though theoretically our mobile robots could turn about

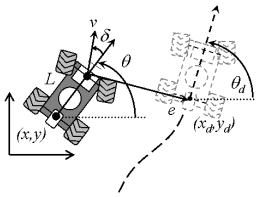

Fig. 4. Path following kinematic control for a car-like vehicle using lateral cross track error from reference path.

a point, this is not true on many high friction surfaces such as grass. We offset the control point from the center of gravity of the robot by a distance L. For the robot to track a reference curve (path), we search for a reference point on the curve that is closest along the normal to the current robot heading direction (see Figure 4) using a *lookahead* distance along the reference curve.

We use a nonlinear steering controller [19] for path following using cross track error $e(t)$ as feedback, with some modifications due to the absence of steering wheels on our robot. Further, we ignore the effects of car-like dynamics while tracking curves and drop the non zero steady state yaw error. If $v(t)$ is the forward speed of the robot and $\psi(t) = (\theta_d(t) - \theta(t))$ is the heading with respect to the tangent at the closest trajectory point (Figure 4), for global asymptotic convergence to zero cross track error we use the control law: $\delta(t) = \psi(t) + \arctan\left(\frac{ke(t)}{v(t)}\right) + k_{steer}\left(\delta_{meas}(i) - \delta_{meas}(i+1)\right)$, with a saturation limit of $\pm\delta_{max}$. k is a proportional steering gain and k_{steer} is a gain modeling the delay in the steering actuation. Note that we map the joystick control inputs to approximate measured effective steering angles δ_{meas} using a lookup table. However we get actual measures of $v(t)$ and $\theta(t)$ from visual odometry running live on the robot. Our spline representation allows us to quickly compute tangents to the reference curve at chosen points and hence $\theta_d(t)$.

The controller compensates for delays in the low level actuators using the last term in the above equation by checking the steering inputs. We also use a function to slow down the robot on tight curves to reduce slip and transient tracking errors. 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. For retro traverse the controller performed adequately (maximum deviation less than half the robot width) at speeds up to $1.2m/s$ with appropriate proportional gains.

B. Obstacle avoidance

For safe and reliable operation of the autonomous path follower we propose two levels of obstacle detection and collision avoidance strategies (shown in Figure 5). The first level is the dynamic *hard stop* which is designed to bring the robot to an immediate stop if a large obstacle suddenly appears in its forward path at a close distance ($< 2.5m$). The second level is *reactive obstacle avoidance* which modifies the follower path locally based on a traversability map of obstacle free areas up to $5m$ away.

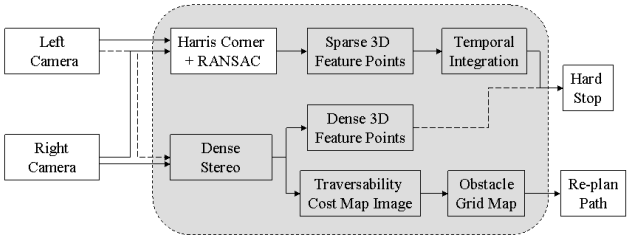

Fig. 5. Flow diagram for obstacle detection using sparse and dense features from a stereo pair. The Harris corner features (outliers removed) are already available from the visual odometry module so the hard stop detection happens in the order of $1ms$ after a frame is processes by visual odometry. To keep the overall frame rate at $10Hz$ we subsample the input images by a factor of 4 before the dense multi-resolution correlation based stereo computation ([21]). This gives us enough resolution to do obstacle detection out to $6-8m$.

As described earlier, our visual odometry algorithm generates a set of sparse 3D feature points based on image Harris corners detected and stereo matching. The hard stop obstacle detection algorithm fills in a ground aligned Cartesian grid where each cell stores the height above ground for 3D points detected at that location. It then aggregates the height statistics for each grid cell within the designated area of interest (near robot, in its forward path) and applies thresholds for acceptable sizes of obstacles after some noise removal. Using this information, the hard stop module reports the presence of lethal obstacles within the zone being monitored. We arrived at thresholds for acceptable levels of performance after extensive in-field trials. Some detection results and corresponding sparse feature points are shown in Figure 6.

Our obstacle detector [21] determines the location and severity of obstacles in a scene described by 3D range data. The range data is derived from a real-time dense multi-resolution correlation based stereo algorithm and the camera pose comes from the visual navigation solution. The algorithm produces an analog output that measures obstacle severity (in terms of height, slope and size), so that subsequent thresholding can determine which obstacles are considered to be traversable, based on the characteristics of the vehicle. Frame-level obstacle maps are then stitched together, in real-time, to form a local traversability map that is provided as input to the planner. The algorithm proceeds as follows. The stereo range data is pre-filtered to reduce noise and then transferred to Cartesian ground coordinates from image coordinates such that ground resolution variation is accounted for. The data is then analyzed at multiple resolutions to detect obstacles with varying slopes. The obstacle height is computed from the heights of points on the obstacle boundary. The map representing these heights is merged with the slope contribution to yield a traversability cost map (see Figure 7). The reactive path planner uses a time aggregated version of this map to navigate around obstacles locally.

We represent the accumulated traversability cost map (within a local window of $5m$ radius) as an obstacle *gridmap* image with each pixel representing a grid cell and compute its distance transform using a pseudo Euclidean distance metric for speed. Next, a clearance distance threshold is

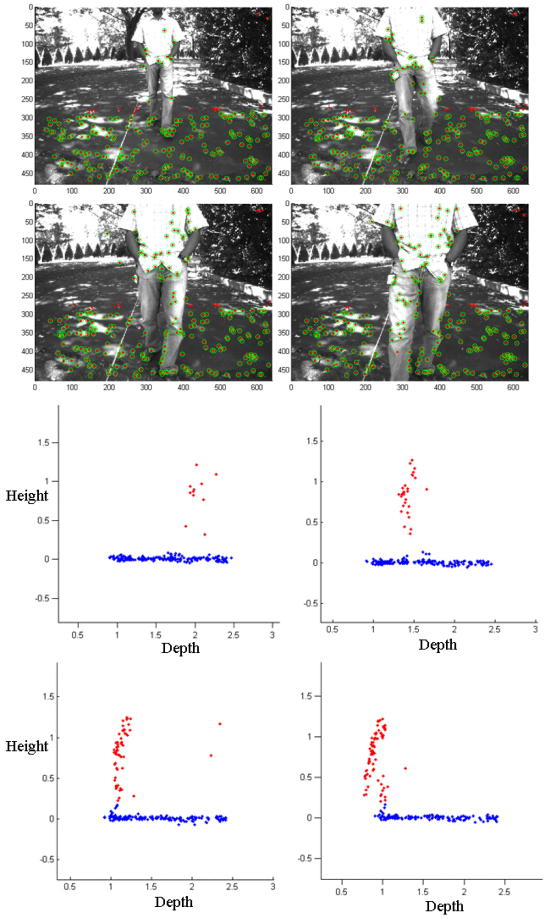

Fig. 6. Obstacle detection for hard stop using sparse 3D features. The top set of four images shows the Harris corner point features which satisfy the inlier and stereo matching criteria in red. The points that are closer than 2.5m and used by the obstacle detector are circled in green. The bottom set of plots show the ground plane points in blue and the obstacle points in red.

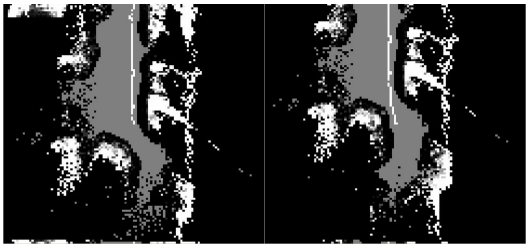

Fig. 7. Obstacle traversibilty maps based on dense stereo for a typical outdoor run. The black regions are unexplored or occluded, the gray region are best areas to traverse while lighter colors are increasingly hazardous with white being lethal.

applied to the output of the distance transform based on the robot's geometry and maneuverability. This has the effect of growing the obstacle boundaries to accommodate for the finite robot size. Figure 8 shows this procedure for one such planning step. Note that the robot is at the center of the map here with the heading vector as shown. After the above step the reactive planner checks the reference path to see if it will intersect any obstacles within the local window. When collision with obstacles is locally imminent, we coarsely sample the reference path ahead, and assuming the robot heading to be the local path tangents at those points, we compute a polar histogram (angles as bins) of

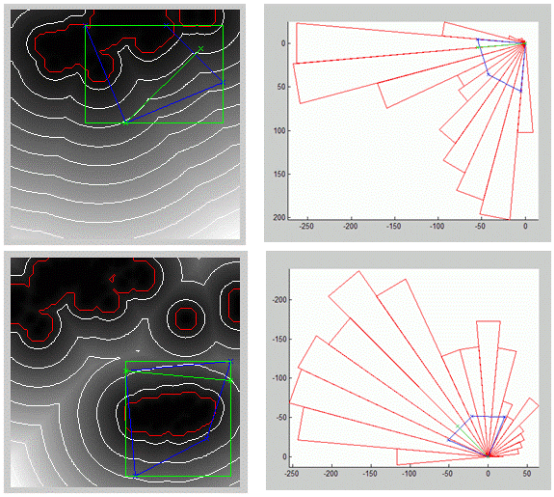

Fig. 8. (Left) Distance transform of obstacle gridmap image with contours drawn for easier visualization. The red contour marks the clearance threshold around obstacles. The blue lines show the current field of view approximation and the green line shows the histogram mode direction. (Right) Angular clearance histogram showing distribution of obstacle free points within specified field of view for the images on the left. Green line shows mode sector. The angular orientations are not aligned with the plots on the left.

obstacle free pixels within a specified field of view (Figure 8) at each point. The mode of the histogram locally gives the "best" obstacle free angular sector for that point. We use this angular direction (from current heading) to search for a sector with obstacle clearance above a minimum threshold. We then find the closest point on the reference path along this vector, while modifying the local path to go along this vector from the previous point. This process is repeated until the forward path is locally free of obstacles. We are currently investigating more optimal re-planning strategies.

IV. System integration and testing

In the ViewTrek system shown in Figure 1, the leader and the follower use similar sensors. The hardware details are provided in [20], we briefly highlight some key parts. The follower system communicated with the robot via a wired serial interface. In both the helmet and the robot systems the cameras point toward the ground at an angle of approximately 15 degrees from horizontal in order to capture nearby features. Since the helmet-based leader system needs to keep up with fast motions, its dead reckoning navigation was operating at 15 Hz, while the slower moving robot system operated at 10Hz, with the extra processing time used by the dense stereo based obstacle detection and reactive planning. Both systems generate, wirelessly transfer and match landmarks asynchronously at 1Hz. The average bandwidth use for our system is about 100KB/s.

A. System operation

The ViewTrek system was field-tested for autonomous retro-traverse as well as leader follower modes. For retro-traverse, the operation proceeded as follows: (1) The operator turns on robot and waits for visual navigation system to initialize. (2) The operator manually drives robot using

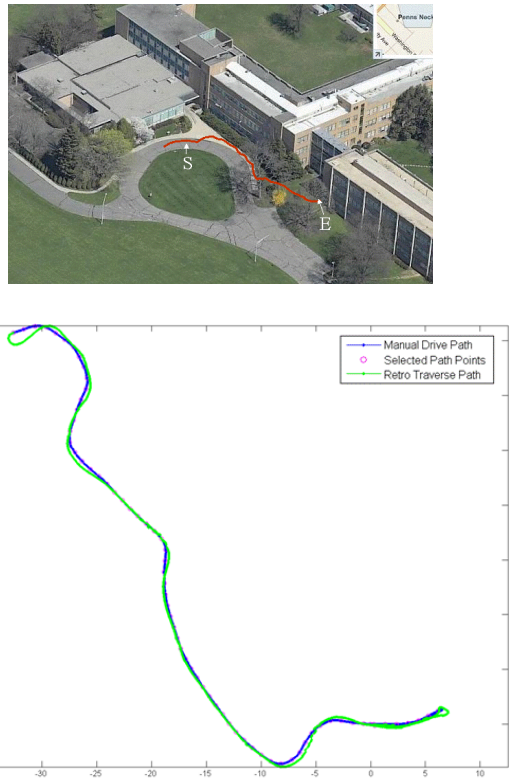

Fig. 9. Results of a typical retro-traverse run showing ground aligned 3 DOF pose derived from 6 DOF visual navigation pose output. The start (and return) position of the robot is at the origin for each plot. Note the path goes along a narrow sidewalk and over grass through bushes on either side. We have achieved similar results indoors in corridors as well.

joystick control, selects a starting "home" position for the robot and continues driving. (3) Once the robot has reached its final destination, the operator initiates retro-traverse by pressing a joystick button. (4) The robot pauses and then turns around and starts following its path back to the home base.

For the helmet leader-follower system the operation proceeds as follows: (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 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 follows). (6) Upon reaching the destination, the operator deactivates the system via the tablet computer interface.

At the beginning of each leader following run, the operator wearing the helmet should stand alongside the robot for initial synchronization. This makes it easy to overlap the fields of view of the robot and helmet. The synchronization

TABLE I

EVALUATION OF ACCURACY FOR VISUAL ODOMETRY (VO), LANDMARK MATCHING (LM) AND THE COMBINED 3D POSITION (VO+LM).

Min (m)	Max (m)	Median (m)	Mean (m)	Length (m)
				111.9
0.00089	0.69675	0.08804	0.14325	VO
0.00002	0.23349	0.01296	0.01973	LM
0.00028	0.39436	0.04246	0.05718	VO+LM
				253.66
0.000136	1.06107	0.33095	0.36808	VO
0.00003	0.29118	0.02032	0.03025	LM
0.00014	0.52447	0.05307	0.08575	VO+LM
				266.62
0.00027	0.91672	0.32611	0.34266	VO
0.00020	0.31580	0.01716	0.02550	LM
0.00142	0.34655	0.07399	0.09577	VO+LM

usually happens within a few seconds following system activation. In both autonomous path following modes the obstacle detection and avoidance can be activated. Further, the operator can take over control at any point using the joystick, drive the robot a short distance and then resume path following for the remaining path. The operator can choose the path following speed and standoff distances via configuration files before the start of each run.

B. Results

We first experimentally tested the accuracy of the pose output from each component of our visual navigation system under controlled conditions, followed by closed loop testing of the entire ViewTrek path follower system in various outdoor environments.

1) Visual Odometry and Landmark Matching Accuracy: In order to evaluate the accuracy of the visual odometry and landmark matching modules, a set of experiments are performed on the mobile robot. The ground truth positions are obtained via a Leica *Total Station* surveying device with automatic tracking capability. Specifically, a prism is installed on the top of the visual sensor rig and the relative pose between the prism and the visual sensor is calibrated. During these experiments, the operator drove the robot along a set of predefined paths. Along each path, the Total Station device tracks the prism automatically to obtain its 3D position while the robot is moving. Meanwhile, the 3D positions from the visual navigation system are also recorded. Since the visual sensor rig is synchronized with the Total Station device, each location has a pair of total-station and visual navigation system position measurements.

Once we collected the synchronized position measurements for all the paths, the accuracy of both visual odometry and landmark matching could be evaluated. For the visual odometry accuracy, the error was computed directly as the difference between the total-station position measurements and the visual sensor position measurements at each location. For the landmark matching accuracy, the accuracy was computed for a location only when there was a successful landmark matching at that location. Specifically, the distance between the reference landmark location and the current (matched) landmark location was measured independently for both the total-station and the visual sensor. The difference

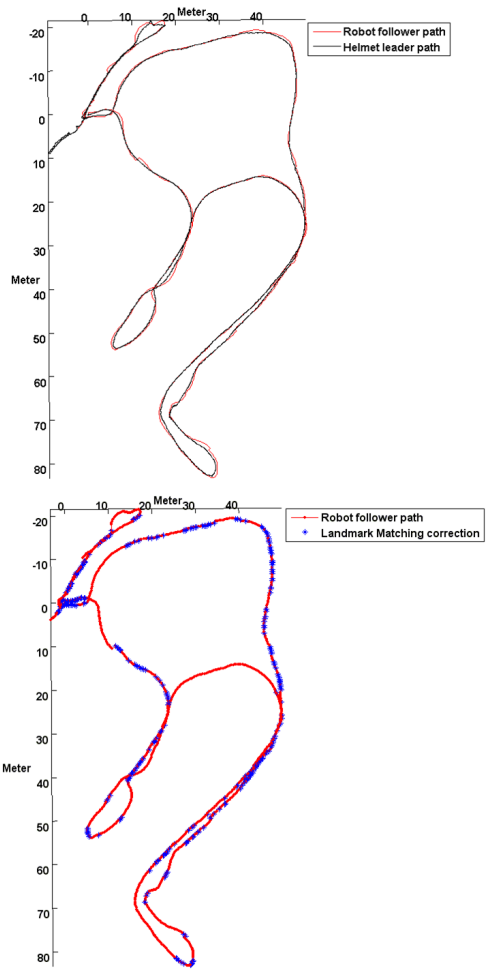

Fig. 10. Leader and follower paths for a typical run. Landmark match and hence pose correction locations shown on follower's path - note the jumps when there is a match after a longer interval.

between these two distances is defined as the landmark matching error. As shown in Table I, without the landmark matching, the average drift rate of the system with visual odometry (multi-camera and IMU) system alone was found to be less than 0.15% of travelled distance. The average landmark matching error was found to be around $3cm$.

2) Field tests for retro-traverse and leader-follower operation modes: The ViewTrek system has undergone 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. We performed retro-traversal tests both indoors and outdoors along narrow passages, including a week long testing involving rough terrain at a military location. A sample run at our suburban campus is shown in Figure 9. One major challenge for the leader-follower mode of the system 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

TABLE II

EXPERIMENTAL RESULTS FOR VARIOUS PATH FOLLOWER RUNS.

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

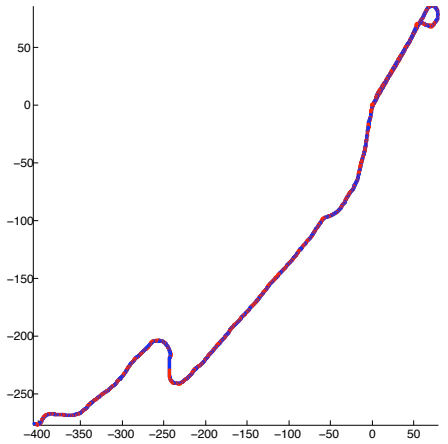

Fig. 11. Path of leader (blue) and follower (red) over a long run of $868m$.

operator.

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 to the side), the dead-reckoning visual odometry system allows the robot to follow for 10s of meters (see [4] for a quantitative evaluation) without registering a landmark match. Figure 10 shows the leader and follower paths highlighting the locations on the follower's path where landmark matching occurs and the pose correction jumps.

Various runs, with several different operators and including ones in a suburban environment (see Figure 9) are shown in Table II. The last 2 runs were in a desert environment. The longest recorded run of $868m$ is shown in 11. These results clearly show robustness and consistent performance under a variety of circumstances.

V. CONCLUSIONS

We presented a system capable of autonomous following and retro-traverse maneuvers. 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 follower robot must only maintain its pose with respect to the leader's traversed path, not the global coordinate system. Similarly, for a robot doing retro-traverse, pose has to be maintained with respect to the starting (home base) position. 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 leader-follower application is in the live, automatic sharing of visual landmarks between the operator wearing a sensor-equipped helmet and an autonomous robot. This technology can also be used to lead convoys of autonomous vehicles, each following its leader's landmark trail.

REFERENCES

[1] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," *Computer Vision and Pattern Recognition, IEEE Conf. on*, Jan 2004.

[2] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *Journal of Field Robotics*, vol. 24, no. 3, 2007.

[3] M. Bansal, A. Das, G. Kreutzer, J. Eledath, R. Kumar, and H. Sawhney, "Vision-based perception for autonomous urban navigation," *Intelligent Transportation Systems, 11th International IEEE Conf. on*, 2008.

[4] 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, IEEE Conf. on*, May 2007.

[5] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," *Intelligent Robots and Systems, IEEE/RSJ Intl. Conf. on*, Jan 2008.

[6] M. Agrawal and K. Konolige, "Real-time localization in outdoor environments using stereo vision and inexpensive gps," *Intl. Conf. on Pattern Recognition*, Jan 2006.

[7] C. Engels, H. Stewenius, and D. Nister, "Bundle adjustment rules," *Photogrammetric Computer Vision*, Sept.

[8] K. Konolige and M. Agrawal, "Frameslam: From bundle adjustment to real-time visual mapping," *Robotics, IEEE Trans. on*, vol. 24, no. 5, Oct 2008.

[9] G. Sibley, C. Mei, I. Reid, and P. Newman, "Adaptive relative bundle adjustment," in *Robotics Science and Systems Conference*, 2009.

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

[11] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," *Computer Vision and Pattern Recognition, IEEE Conf. on*, 2006.

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

[13] 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, IEEE Conf. on*, May 2008.

[14] M. Cummins and P. Newman, "Accelerated appearance-only SLAM," in *Robotics and Automation, IEEE Intl. Conf. on*, Pasadena,California, April 2008.

[15] T. Goedeme, T. Tuytelaars, L. V. Gool, G. Vanacker, and M. Nuttin, "Feature based omnidirectional sparse visual path following," *Intelligent Robots and Systems, IEEE/RSJ Intl. Conf. on*, Jul 2005.

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

[17] A. Diosi, A. Remazeilles, S. Segvic, and F. Chaumette, "Outdoor visual path following experiments," in *Intelligent Robots and Systems, IEEE/RSJ Intl. Conf. on*, Nov. 2007.

[18] S. Thrun, "Winning the darpa grand challenge: A robot race through the mojave desert," in *Automated Software Engineering, 21st IEEE/ACM Intl. Conf. on*, Sept. 2006.

[19] 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 Conf.*, Jun 2007.

[20] O. Naroditsky, Z. Zhu, A. Das, S. Samarasekera, T. Oskiper, and R. Kumar, "Videotrek: A vision system for a tag-along robot," in *Computer Vision and Pattern Recognition, IEEE Conference on*, June 2009.

[21] A. Rieder, B. Southall, G. Salgian, R. Mandelbaum, H. Herman, P. Rander, and T. Stentz, "Stereo perception on an off-road vehicle," in *IEEE Intelligent Vehicle Symposium*, June 2002.