I Can See for Miles and Miles: An Extended Field Test of Visual Teach and Repeat 2.0

Michael Paton, Kirk MacTavish, Laszlo-Peter Berczi, Sebastian Kai van Es and Timothy D. Barfoot

Abstract Autonomous path-following systems based on the Teach and Repeat paradigm allow robots to traverse extensive networks of manually driven paths using on-board sensors. These methods are well suited for applications that involve repeated traversals of constrained paths such as factory floors, orchards, and mines. In order for path-following systems to be viable for these applications they must be able to navigate large distances over long time periods, a challenging task for vision-based systems that are susceptible to appearance change. This paper details Visual Teach and Repeat 2.0, a vision-based path-following system capable of safe, long-term navigation over large-scale networks of connected paths in unstructured, outdoor environments. These tasks are achieved through the use of a suite of novel, multi-experience, vision-based navigation algorithms. We have validated our system experimentally through an eleven-day field test in an untended gravel pit in Sudbury, Canada, where we incrementally built and autonomously traversed a 5 Km network of paths. Over the span of the field test, the robot logged over 140 Km of autonomous driving with an autonomy rate of 99.6%, despite experiencing significant appearance change due to lighting and weather, including driving at night using headlights.

1 Introduction

Autonomous path-following algorithms based on the Teach and Repeat paradigm allow robots to repeat networks of connected paths previously driven by human operators using only on-board sensors.

The unique task of autonomously traversing a human-taught path gives a robot a strong prior on where it is safe to drive [2]. This allows for confident, autonomous navigation through rough, outdoor terrain that would otherwise be inaccessible or

M. Paton (✉) · K. MacTavish · L. -P. Berczi · S. K. van Es · T. D. Barfoot
University of Toronto Institute for Aerospace Studies,
Toronto, ON M3H 5T6, Canada
e-mail: mpaton@robotics.utias.utoronto.ca

T. D. Barfoot
e-mail: tim.barfoot@utoronto.ca

M. Hutter and R. Siegwart (eds.), *Field and Service Robotics*, Springer Proceedings
in Advanced Robotics 5, https://doi.org/10.1007/978-3-319-67361-5_27

require complex, generic, and potentially risky terrain-assessment algorithms. Furthermore, these methods can be implemented to have bounded computation costs and minimal map sizes [7], making them well suited for long-range navigation. These benefits make autonomous path following appealing for industrial applications that consist of repeated traversals over constrained paths, such as factory floors, orchards, and mines. They are also well suited to applications that consist of autonomous exploration and retrotraverse such as search-and-rescue and hazardous-exploration robots. However, autonomous path-following systems suited for these applications need the ability to navigate large-scale environments over long time periods. Furthermore, they require constant metric localization to the manually driven path as input to a path-tracking controller to ensure minimal drift, and the ability to recognize and cope with obstacles blocking the path. These requirements pose a serious challenge for vision-based systems whose advantages of cost and commercial ubiquity come at the expense of robustness to appearance change. This paper presents Visual Teach and Repeat (VT&R) 2.0, a path-following system capable of long-term, navigation on large-scale networks of paths using only a stereo camera through the integration of a suite of recently published navigation algorithms [2, 14, 19, 21]. Furthermore, we present results from an extensive outdoor field test, illustrated in Fig. 1, that consisted of incrementally building and autonomously traversing a 5 km network of connected paths over the span of eleven days at an untended gravel pit in Sudbury, Canada. Over these eleven days, the robot traversed over 140 km with an autonomy rate of 99.6% of distance traveled while experiencing significant appearance change due to lighting, weather, and terrain modification.

The remainder of this paper is outlined as follows. Work related to VT&R 2.0 is summarized in Sect. 2. Details of the VT&R 2.0 system are presented in Sect. 3.

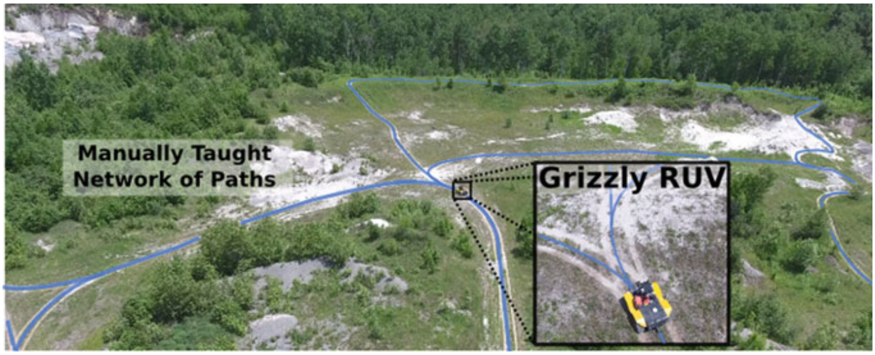


Fig. 1 A Grizzly RUV deployed with an autonomous path-following algorithm navigating a 5 km network of manually taught paths. Applications that rely on repeated traversals of constrained paths will greatly benefit from such algorithms: such as mining, agriculture, and patrol robots. In order to be useful for such applications, autonomous path-following algorithms will need to be able to cope with large-scale maps, and appearance change over long periods of time. Using a novel multi-experience localization and mapping method, the robot pictured above autonomously traversed over 140 km over two weeks, experiencing significant appearance changes in the environment

Section 4 provides information on the experimental set up of the field test with results presented in Sect. 5. Failure conditions of VT&R 2.0 and lessons learned from the field are presented in Sect. 6 with a conclusion in Sect. 7.

2 Related Work

VT&R 2.0 is an evolution of our previous system that provides short-term, vision-based path following on large-scale tree structures, VT&R 1.0 [6, 7]. While effective at long-range navigation, the system is highly susceptible to lighting change while operating outdoors, limiting successful operation to a window of only a few hours. This method was extended to multi-day operation through the use of color-constant images and multiple stereo cameras [18], but is still susceptible to longer-term appearance change due to weather and seasons. Apart from VT&R 1.0, short-term, vision-based path-following systems have been demonstrated using heading-only navigation [4, 10]. Despite the fusion of wheel odometry, the lack of a reliable translation estimate makes navigation in constrained environments unsafe. Path-following systems that rely on active sensors provide long-term, lighting-invariant navigation and have been demonstrated using appearance-based methods on lidar-generated intensity images [15] and point-cloud registration on dense 3D scans [11]. However, these systems struggle with motion distortion issues and rely on expensive and sometimes commercially unavailable sensors.

It is well known that constant-time localization and mapping can be achieved using globally inconsistent, topometric pose graphs [20], even with loop closures. Long-range navigation with VT&R 2.0 is possible through the use of this representation, allowing the system to build large-scale networks of paths with smooth path tracking across loop closures [21]. Long-term navigation with VT&R 2.0 is achieved through the use of the Multi-Experience Localization (MEL) algorithm [19]. This method is inspired by the Experience-Based Navigation (EBN) framework [5], which localizes against a number of past experiences, providing metric localization to the most similar experience. In contrast, MEL provides localization to the *single* manually taught path using experiences gathered *during* autonomous operation. Due to their reliance on multiple experiences, both EBN and MEL are inherently computationally intractable if left unbounded. A viable strategy to overcome this issue is to select a fixed-size subset of experiences for the localizer. [12] use past localization success to recommend experiences most likely to localize well in the future for the EBN system. In contrast, our system selects experiences most similar visually to the *live view* [14]. Another work closely related to MEL is the ‘Summary Maps’ method [16]. This method provides metric localization across seasonal appearance change through a multi-experience map that is pruned and curated offline. This method is successful, but requires downtime between traverses to perform mapping on an offline server—a constraint which restricts use for some applications.

Safe navigation with VT&R 2.0 is achieved through a place-dependent, multi-experience terrain-assessment algorithm [2]. Traditional terrain-assessment methods are limited by the ability to label the terrain in a human-interpretable way, often resulting in conservative estimates [8]. More recent methods alleviate these issues through the use of previous experiences to learn traversability [3, 9], but still lead to conservative estimates in difficult environments. In contrast, our system learns a separate classifier at every place on the path [2], achieving better performance than a single general classifier applied to every place.

3 System Overview

This section provides an overview of the following components of VT&R 2.0: (i) map structure, (ii) network construction, (iii) route planning, (iv) autonomous path following, (v) multi-experience localization, (vi) appearance-based experience selection, and vii) place-dependent terrain assessment.

The Spatio-Temporal Pose Graph: Our system stores the map in a database indexed by a Spatio-Temporal Pose Graph (STPG). Depicted in Fig. 2, the graph structure contains vertices, temporal edges, and spatial edges. Vertices, each with a reference frame, $\underrightarrow{\mathcal{F}}$, store raw sensor observations and triangulated 3D landmarks with associated covariances and descriptors.[1] The edges link vertices with uncertain, relative $SE(3)$ transformations. Temporal edges (blue lines) connect temporally adjacent vertices, while spatial edges (green lines) connect those that are spatially close (but may be temporally distant). Edges are considered *privileged* (solid lines) if the robot was being manually driven, or *autonomous* (dashed lines) if the robot was autonomously repeating a route. VT&R 2.0 uses the STPG to represent a multi-experience network of connected paths, where each *experience* is a collection of vertices linked by temporal edges. The subgraph containing *all* privileged experiences represents the collection of safe, drivable paths. Autonomous experiences linked to this privileged subgraph are used to aid the navigation algorithms, by providing a wealth of place-specific information.

Network Construction: VT&R 2.0 is operated through the user interface shown in Fig. 3, which provides an intuitive means to construct networks of paths, and command autonomous traversal to goals on the networks. A network is built by adding a teach goal (left panel) and manually driving the robot; this adds privileged experiences to the STPG. If the network exists prior to the teach, then a localization search centered around the robot's topological state estimate is performed. Upon successful localization, a privileged spatial edge is created, *branching* the new experience off the existing network. The robot will then add vertices and temporal edges to this new experience through our stereo VO pipeline [19] as it drives. Live experiences can be merged back into the network through loop closures initiated through the UI. The operator selects a region for the merge, and the system attempts to localize the

[1] We use SURF features triangulated from greyscale and color-constant stereo measurements in our implementation, but the overall system is generic to any point-based, sparse visual feature.

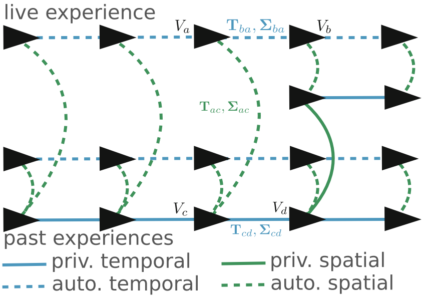
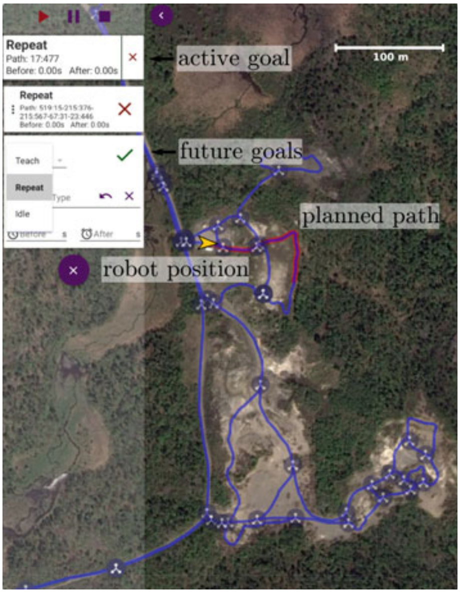


Fig. 2 Overview of the STPG data structure used to represent our multi-experience network of paths. Experiences are shown as rows of vertices (black triangles) connected metrically through blue temporal edges calculated via VO while the robot is either being manually driven (solid) or autonomously repeating (dashed). Experiences are related metrically through green, spatial edges, calculated through localization and can either be added autonomously while driving (dashed) or manually while adding a branch or loop closure (solid)


Repeat
Path: 17:477
Before: 0.00s After: 0.00s

Repeat
Path: 519:15-215:376-215:567-67:31-23:446
Before: 0.00s After: 0.00s

Type: Repeat

Fig. 3 Overview of the VT&R 2.0 user interface used to build networks of connected paths and command the rover to autonomously traverse the network

live view to the privileged vertices in this selection. Upon successful localization, the operator can confirm the result, adding a privileged spatial edge from the live vertex to the target. Otherwise, the user may continue driving to improve alignment, or cancel the merge.

Route Planning: Autonomous traversal begins with planning a route in the UI by adding a repeat goal (active in Fig. 3) to the queue, and selecting a sequence of waypoints for the robot. To plan the path, the system uses the safe, privileged subgraph of the network, including privileged experiences, and privileged spatial edges from branching and merging. Given the robot's current topological position in the network and a set of waypoints, the planner finds the minimum-cost path that covers all selected waypoints in sequential order. Two edge costs we find useful are the path distance, and the temporal age between the live experience and the most recent traversal of the edge (combining absolute time and time-of-day). Since our system localizes against multiple autonomous experiences, planning over recently traversed edges is a heuristic to improve the likelihood of successful localization.

Path Following: Given a planned route through the privileged network, the system creates a new autonomous experience in the network, and attempts to localize the live view to a vertex in the privileged path. This process is identical to the first step of teaching, except the added spatial edge is flagged as autonomous. Once connected to the privileged experience, the system propagates the position estimate using stereo VO, which is sent to a model-predictive-control path tracker [17]. When a new keyframe is added as a vertex in the live run, it is localized to the closest vertex in the privileged path using MEL, detailed in the next subsection. Upon successful localization, an autonomous spatial edge is added between the two vertices.

Multi-Experience Localization: VT&R 2.0 provides metric localization with respect to the privileged path through the Multi-Experience Localization (MEL) algorithm [19]. Illustrated in Fig. 4, MEL estimates the uncertain transformation, $\{\hat{\mathbf{T}}_{bd}, \hat{\boldsymbol{\Sigma}}_{bd}\}$ (purple, dashed line), between the live vertex, V_b, and the estimated closest privileged vertex, V_d. This process takes a window of recommended experiences (green rectangles) as input, and transforms their landmarks to the coordinate frame

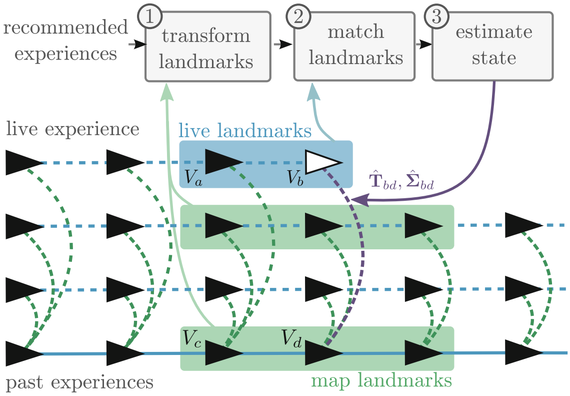


Fig. 4 An overview of the MEL algorithm. Given a selection of experiences to localize against (green rectangles), The algorithm solves for the transformation between the vertex in the live experience, V_b, and the vertex in the map experience, V_d, by transforming all landmarks in the localization window into the coordinate frame of V_d and performing a simple keyframe-to-keyframe bundle adjustment problem

of the target privileged vertex, V_d, using the temporal and spatial edges. The uncertainty of the transformed landmarks includes that from the source landmark and the transform itself. Landmarks originating from the live vertex, V_b, are then matched against these consolidated map landmarks. Matching consists of checking the similarity of unmatched, live landmarks to landmarks at each map vertex, starting at V_d, and expanding in a breadth-first search. Similarity is determined by the keypoint Laplacian sign, feature descriptor distance, and a weak check on projected landmark position in image space using the prior. The process continues until one of the following exit conditions is met: (i) enough matches are found, (ii) the time limit has expired, or (iii) the map window is exhausted. The matches are first verified to remove outliers and initialize $\mathbf{T}_{bd}$, and a motion prior is built by compounding VO transforms and the most recent successful localization. The uncertain transform, $\{\mathbf{T}_{bd}, \mathbf{\Sigma}_{bd}\}$, is iteratively refined in a robust, nonlinear, least-squares optimization using our Simultaneous Trajectory Estimation and Mapping (STEAM) engine [1].

Appearance-Based Experience Selection: Matching against all intermediate landmarks in the MEL algorithm becomes computationally intractable as the number of experiences grows. Only a subset of these are actually required to localize the live vertex—if we restrict our localization problem to the most relevant, we maintain real-time performance. To determine a relevant subset of experiences, we would like to find those with similar appearance to the *live view*. To this end, we have developed an algorithm [14], that selects a small subset of experiences based on a BoW appearance summary, illustrated in Fig. 5.

Each vertex in the STPG contains a BoW descriptor, quantized from the stable (tracked by VO) features in the keyframe against a local visual vocabulary. To improve robustness to viewpoint and signal-to-noise ratio, a grouped descriptor is

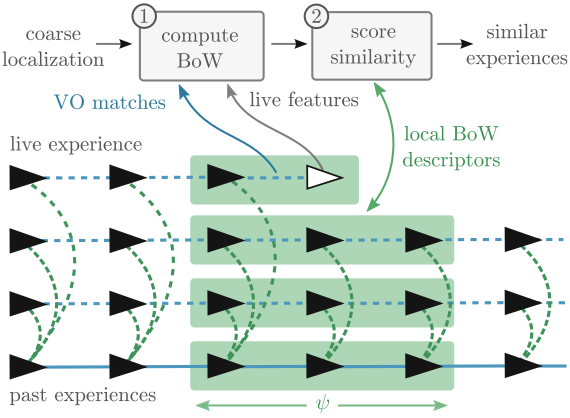


Fig. 5 An overview of the experience selection algorithm using BoW. 1. Given the VO feature matches for the live frame, a sliding-window BoW descriptor is constructed. 2. The BoW descriptor is compared against those for past experiences, centered on the coarse localization. The most similar experiences are selected for use in the remainder of the localization problem

formed by summing the descriptors from a local group of vertices, ψ, from the same experience [13]. At the start of each MEL problem, the local BoW descriptor (green rectangle) around the live vertex (white triangle) is compared to those of each past experience (lower green rectangles) using the cosine similarity. The N experiences with the highest similarity to the live experience are selected for MEL, where N is chosen to maintain real-time performance (5–10). Since this comparison is very fast, this method allows us to triage a large number (100s) of experiences very quickly.

Place-Dependent Terrain Assessment: We exploit the fact that during VT&R, the robot only needs to assess the traversability of terrain that it has already seen and driven. All of the drivable paths were at one point manually taught, and we assume that they were safe at that point in time. Traversability of the terrain ahead of the robot is determined by comparing it to known safe examples from previous experiences, and labeling any terrain that is sufficiently different as unsafe. This reduces the problem of terrain assessment to the simpler problem of change detection—allowing slow, gradual change over the lifetime of the path (such as growing grass), and stopping for large, sudden changes (such as a fallen tree). This place-dependent terrain-assessment algorithm first appeared in [2], and an overview of the pipeline is shown in Fig. 6. Patches are compared by taking the absolute difference of individual cell heights between the lookahead patches and each training patch individually. The patch cost is defined as the worst of these cell differences for each lookahead-training pair, and the traversability of the terrain is determined based on the lowest cost to previous experiences.

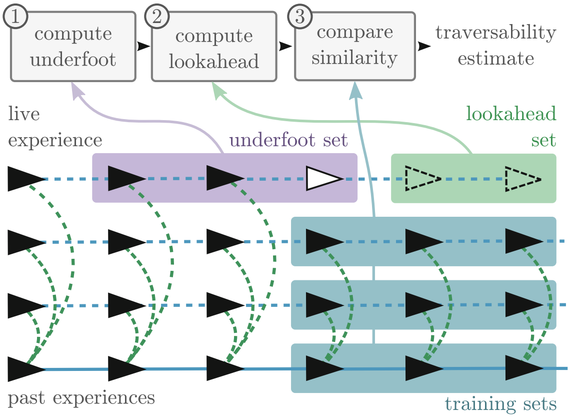


Fig. 6 An overview of the place-dependent terrain assessment. 1. Patches are computed at the robot location using recent data in the *underfoot set* (purple). 2. Patches are computed ahead of the robot at vertices in the *lookahead set* (green). 3. Lookahead patches are compared to previously computed underfoot patches from a spatially local *training set* (blue). The traversability ahead of the robot is based on its similarity to previous experiences

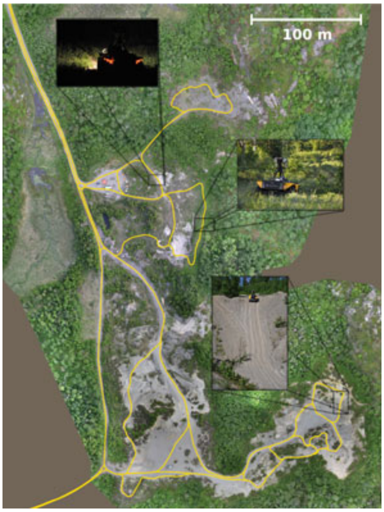


Fig. 7 Orthomosaic imagery of the 5 km network of paths at the Ethier Sand and Gravel in Sudbury, Ontario, Canada

4 Experimental Setup

Between the dates of 10/06/2016 and 16/06/2016, an extended field test of VT&R 2.0 was conducted at an untended gravel pit in Sudbury, Canada. Illustrated in Fig. 7, this field test was designed to stress test our system's ability to traverse a large-scale network of paths, in an unstructured environment, for an extended period of time, over significant appearance change. This location was selected for its variety of challenging environments, including rich vegetation and shifting sand with little visual texture.

The hardware configuration for the this field test consisted of a Clearpath Robotics Grizzly RUV, shown in Figs. 7 and 7, equipped with a Point Grey Research (PGR) Bumblebee XB3 stereo camera, and a pair of 9-watt LED headlights for nighttime operation. All of our VT&R 2.0 code ran on a Lenovo P50 laptop with a Intel® Core$^{\text{TM}}$ i7-6820HQ CPU equipped with a Quadro M2000 GPU.

Daily field test activities are outlined in Table 1. The majority of the network was taught on the first three days of testing during overcast conditions. During the first five days, the network was traversed from day to night, accumulating over 95 km of driving. On each of the remaining six days, the robot autonomously traversed between 5 and 10 km a day. For each day, the autonomy rate remained above 99%. It is interesting to note that the majority of manual interventions occurred on the first two sunny days. Details on the causes of manual interventions are left for Sect. 6.

Table 1 Overview of the 2016 Sudbury Field Test

Date	Start (hh:mm)	End (hh:mm)	Weather	Teach Dist. (km)	Auto. Dist. (km)	Intervention Dist. (m)	Auto. Rate
2016/06/06	11:15	21:19	Rainy	1.56	13.45	12.56	99.91
2016/06/07	05:42	20:42	Overcast	1.88	17.61	8.74	99.95
2016/06/08	07:21	21:10	Rainy	0.83	23.95	72.66	99.70
2016/06/09	06:25	21:23	Sunny	0.03	19.10	148.69	99.22
2016/06/10	07:13	21:17	Sunny	0.33	20.76	171.62	99.17
2016/06/11	07:40	20:46	Sunny	0.22	09.64	41.49	99.57
2016/06/12	09:59	22:35	Sunny	0.20	08.95	20.04	99.78
2016/06/13	10:38	22:45	Sunny	0.00	07.87	27.65	99.65
2016/06/14	07:33	22:50	Sunny	0.00	07.63	53.42	99.30
2016/06/15	12:16	17:57	Sunny	0.00	07.37	2.27	99.97
2016/06/16	09:16	14:57	Sunny	0.00	04.15	2.85	99.93
Total	–	–	–	5.0	140.5	561.99	99.60

5 Results

This section presents the results of our field test. We begin with a detailed look at localization performance for a 240 m section of the network whose appearance is shown in Fig. 8 and path is highlighted in the top half of Fig. 1 as a blue line. The path begins in a meadow with tall grass and rapidly inclines up to a ridge containing thick vegetation bordered by a tree line, finishing with a steep descent into a sandy gravel pit. During the field test, this stretch of the network was autonomously traversed 109 times with only two manual interventions required. We chose to highlight the performance results of this section of the network in particular because it showcases every variety of appearance change seen during this field test and the rich vegetation and tree line make the environment challenging for vision-based navigation.

Figure 9 shows the relationship between the similarity score used by the experience selector to recommended experiences and feature inlier matches for all autonomous traverses of the example path. The plot shows that experiences with higher scores (>0.6) in general have higher feature match counts to each other, validating the selector's ability to select experiences that will provide more matches.

Figure 10 shows localization results for all 109 autonomous traverses of the path. The left- and right- hand plots show the Cumulative Distribution Function (CDF) of the cross-track uncertainty and inlier feature matches, respectively. We define cross-track uncertainty as the one-standard-deviation uncertainty of our lateral translation estimate relative to the privileged path. The plot can be read as "for y % of the traverse, the localizer reported less than x m of cross-track uncertainty". For the majority of traverses, the cross-track uncertainty stayed below 10 cm. The exceptions are three traverses where the uncertainty rose to 5, 10, and 15 cm at the 80% mark. This same trend can be seen in the right-hand feature inlier CDF, which shows that for all but

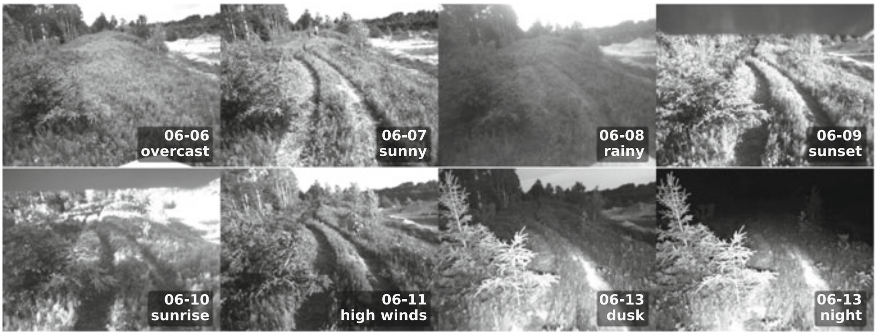


Fig. 8 The changing appearance of the 240 m example path. The top-left image was taken during network construction (no tire tracks), and all subsequent images were successfully localized to it using MEL. This section of the network is challenging for vision-based navigation due to strong shadows cast by a tree line and tall vegetation that moves in the wind

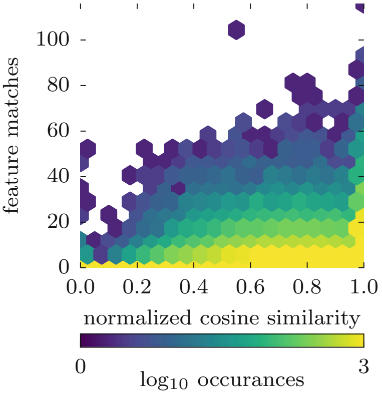


Fig. 9 Experience selector results: Normalized cosine similarity versus feature inlier matches for the vegetation-rich area

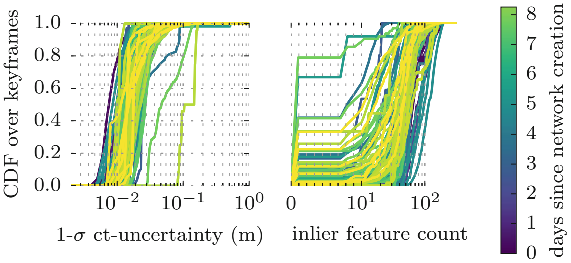


Fig. 10 Localization results for a 240 m section of the network, highlighted in Figs. 1 and 8. *left:* CDF of the $1 - \sigma$ localization uncertainty for all auotnomous traverses on this path. *right*: CDF of the inlier feature matches for all autonomous traverses on this path. *note: log scale on x-axis*

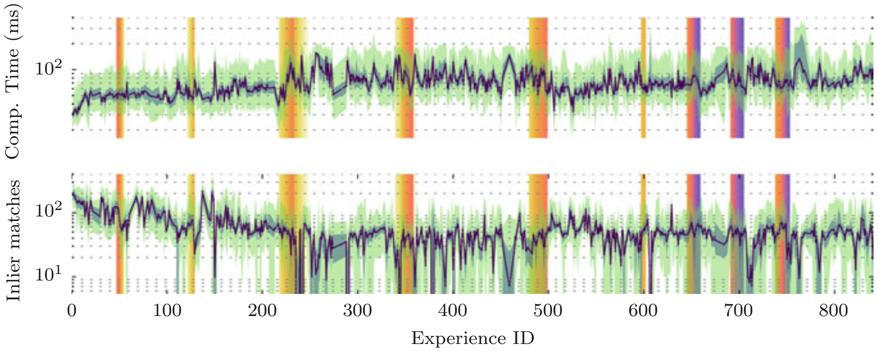


Fig. 11 Localization performance over the entire 11 day field trial with the daylight cycle colored. For each plot the dark line shows the median, the dark shaded area shows the interquartile region, and the light shaded area shows the mini/max extents. *top:* localization computation time of MEL including experience selection. *bottom:* inlier localization feature matches. *note: log scale on y axis*

three runs the localizer had more than 20–30 matches at the 80% mark. This high uncertainty and low match count can be correlated to traverses during sun glare and high winds, causing poor localization performance. More details on these localization failure cases is discussed in Sect. 6.

Figure 11 shows localization computation times and inlier feature matches for every autonomous traverse in the field test. For both plots, the dark-purple line shows the median values of the traverses, the dark-green, shaded area shows the upper (Q2) and lower (Q1) quartile, and the light-green, shaded area shows the min/max inlier values. The background of the plots are colored with respect to daylight conditions. Note the three instances of night driving on the far right. Timing results (top plot) show that the median localization computation time for most traverses is below or near the 100 ms mark. This value is safely within the tolerance for online driving for the VT&R 2.0 system, whose parallelization allows for VO at the frame rate of the sensor and localization at the rate of keyframe creation which is between 2 and 4 Hz.

Feature matches (bottom plot) show that the median and Q1 match count typically stayed between 100 and 30 inlier matches for the majority of traverses. However, there are instances of median values dropping to single digits with Q1 values of zero. These cases can be attributed to navigation in environments challenging for vision-based navigation during difficult weather conditions. These include high winds in vegetation-rich areas, sunshine and terrain modification in open desert areas, strong shadows on tree-lined roads, and glare when the elevation of the sun is low. A detailed analysis of these corner cases can be found in Sect. 6.

During the field tests, the VT&R 2.0 system ensured safe driving through the place-dependent terrain-assessment algorithm. A qualitative example of the algorithm is illustrated in Fig. 12, in which a human blocked the path on the vegetation-rich ridge area illustrated in Figs. 1 and 8. The robot was able to identify a change in the environment caused by the human obstacle and decided to stop. The

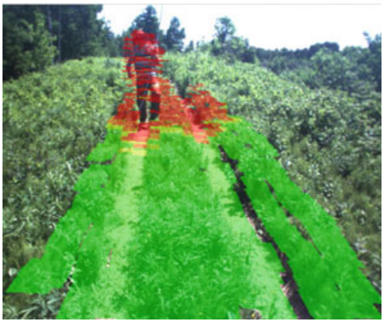


Fig. 12 Example of the place-dependent terrain-assessment algorithm correctly classifiying a human in the path as unsafe

Table 2 Accuracy results for the place-dependent terrain-assessment algorithm.

	Obstacles		No obstacles		
	True Positives	False Negatives	True negatives	False positives	FPR
Count	29	0	20120	56	
Percentage (%)	0.14	0	99.58	0.28	0.28

quantitative performance of the algorithm is measured using the number of true positives (TP) (obstacles correctly identified), false positives (FP) (obstacle detected when no obstacle present), true negatives (TN) (safe terrain correctly identified), false negatives (FN) (missed obstacles), and false-positive rate (FPR), defined as $\frac{\text{FP}}{\text{FP}+\text{TN}}$. which can give more insight into the performance of the system since the nature of the terrain-assessment problem results in many more examples of safe terrain than examples of obstacles.

The terrain-assessment results are shown in Table 2. Notably, there are no false negatives, meaning that all obstacles were identified on the path (100% recall). In the absence of obstacles, only 56 false positives occurred in over 20000 estimates which translates to a false-positive rate of 0.28%. This is extremely low, and is a better indicator of algorithm performance than the typically reported precision (34%) because of the large imbalance of obstacle to obstacle-free examples. The results show that the algorithm is able to safely navigate in challenging environments with very few false positives per distance traveled.

6 Challenges/Lessons Learned

Difficult Conditions: For the majority of the 140 km of autonomous driving, the median inlier match count was at or above a safe value of 50, as shown in Fig. 11.

However, there were a select few traverses with median values near 10 matches, with a Q1 value of zero. This poor localization performance can be attributed to traversing in conditions that are difficult for vision-based navigation. The conditions

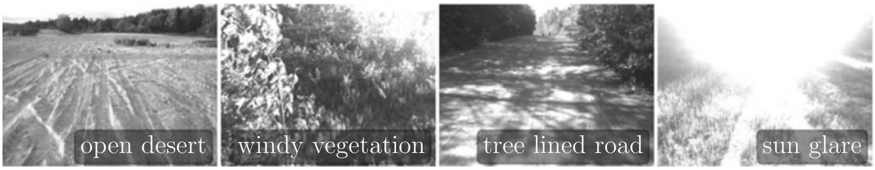


Fig. 13 Example images of areas that remain difficult for vision-based navigation. from left to right: *i* open desert areas with heavy vehicle traffic, *ii* lush vegetation during high winds, *iii* tree-lined corridors with strong shadows, and *iv* sun glare

that were encountered during the field test are highlighted in Fig. 13. Open, desert areas contain few features that persist over time with vehicles, wind, and weather changing the shape of the sand on a daily basis, which causes any stable features to be limited to the horizon. Vegetation-rich areas are typically not a problem for the VT&R 2.0 system, except when high winds are present, which causes the vegetation to rapidly sway back and forth, causing issues for outlier rejection for both localization and VO. Tree-lined corridors cast strong shadows on the road on sunny days. In these conditions, the majority of inlier matches originate from these ephemeral features. With multi-experience systems, this can lead to incorrect state estimates if the majority of inlier matches arise from features that have all moved with the elevation of the sun. The final difficult condition encountered during the field trial is glare from when the sun is low on the horizon, causing images to be oversaturated. It was in these conditions on the sunniest days of the field test where the majority of manual interventions occurred.

Manual Interventions: Figure 14 displays the manual interventions encountered due to localization failures. Trivial interventions were manifestations of a software bug that occurred when the system experienced a VO error, which would stop the robot until the operator drove it forward approximately 0.3 m to trigger a new vertex in the graph. As the nature of this error is somewhat random, the distribution of trivial interventions is nearly uniform across the network. This bug was a minor issue related to the logic of switching between states after a VO error and was resolved after the conclusion of this field test.

Minor interventions were the result of the robot being slightly out of its tracks, which required a small course correction to continue autonomous operation. This was a result of poor localization for short distances, which corresponded to the difficult conditions previously mentioned. Major interventions (yellow lines) occurred when the robot was significantly off course and could not recover. These are a result of extended localization failures without the ability to recover. During the field test the robot experienced interventions in the following areas: i) the tree-lined road when it was sunny and windy (top left), ii) the vegetation-rich area during high winds (mid right), and iii) the open desert area in sunny conditions after heavy vehicle traffic.

A more detailed look at localization performance in this desert area is displayed in Fig. 15. This 50 m path was the most difficult to localize against, running through flat, sandy ground with daily vehicle traffic. After the fifth day of testing, a new path

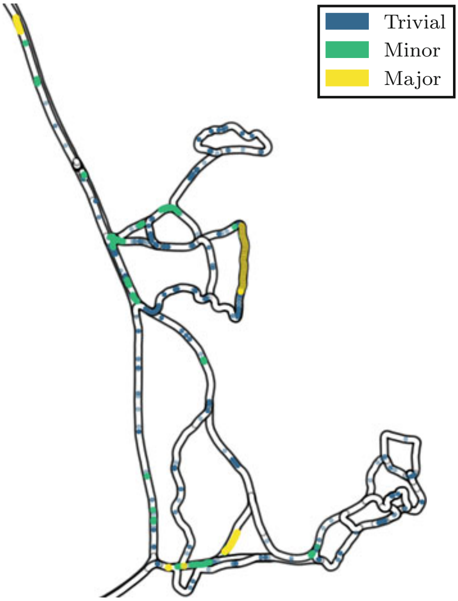


Fig. 14 Manual interventions during the 140 km field trial due to localization issues

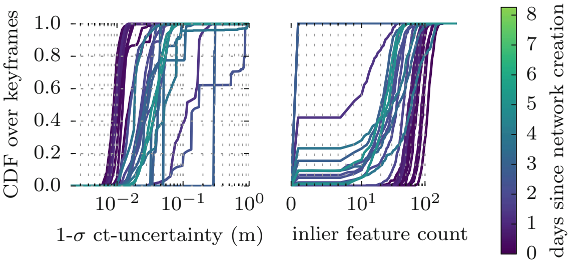


Fig. 15 Localization results a 50 m, open desert section of the network, highlighted in Fig. 13. *left:* CDF of the $1-\sigma$ localization uncertainty for all auotnomous traverses on this path. *right:* CDF of the inlier feature matches for all autonomous traverses on this path. *note: log scale on x-axis*

was rerouted around the trouble area. It can be seen in the figure that the uncertainty is much higher than the vegetation-rich ridge area highlighted in Sect. 5, with $1-\sigma$ cross-track uncertainty between 5 and 10 cm for the majority of the traverses. for three traverses, the maximum uncertainty was as high as 1 m. This corresponds with the feature inlier count, where 40% of the traverse had less than 10 inlier feature matches, for traverses older than 2 days since map creation.

7 Conclusion/Future-Work

This paper presents the long-term, long-range autonomous path-following system, Visual Teach and Repeat (VT&R) 2.0. We present results from an extended field test demonstrating our system's ability to safely traverse large-scale networks of paths across appearance change as different as night-vs-day in challenging, unstructured environments. Over an eleven day period our system traversed a 5 km network of paths accumulating over 140 km of autonomous driving with an autonomy rate of 99.6%. However, there remain challenges for vision-based, autonomous path-following systems related to difficult outdoor conditions that must be addressed. Chief amongst them are environments and conditions with ephemeral ground features such as open deserts, tree-lined roads, and high winds.

Future work on the VT&R 2.0 system will be focused on quantifying the scale of our uncertainty, which is calculated at every stage of the localization process, but has not undergone a rigorous evaluation with respect to ground truth to ensure consistency. Once this process is undertaken, the uncertainty can be used to inform the robot to abandon an autonomous traversal *before* it is too far off the path. Because the VT&R 2.0 system is always adding a new experience into the map, in the event of localization failure, it should be possible to use the current experience to backtrack to a safe area and replan or reattempt the traversal.

Acknowledgements This work was supported financially and in-kind by Clearpath Robotics and the Natural Sciences and Engineering Research Council (NSERC) through the NSERC Canadian Field Robotics Network (NCFRN). The authors would like to also extended their deepest thanks to Ethier Sand and Gravel for allowing us to conduct our field test at their site.

References

1. Anderson S., Barfoot, T.D.: Full steam ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on se(3). In: IROS (2015)
2. Berczi, L.-P., Barfoot, T.D.: It's like déjà vu all over again: Learning place-dependent terrain assessment for visual teach and repeat. In: IROS (2016)
3. Berczi, L.-P., Posner, I., Barfoot, T.D.: Learning to assess terrain from human demonstration using an introspective gaussian-process classifier. In: ICRA (2015)
4. Chen, Z., Birchfield, S.T.: Qualitative vision-based path following. IEEE Trans. Robot. **25**(3), 749–754 (2009)
5. Churchill, W.S., Newman, P.: Experience-based navigation for long-term localisation. IJRR **32**(14), 1645–1661 (2013)
6. Clement, L., Kelly, J., Barfoot, T.D.: Robust monocular visual teach and repeat aided by local ground planarity and color-constant imagery. JFR **34**(1), 74–97 (2017)
7. Furgale, P., Barfoot, T.D.: Visual teach and repeat for long-range rover autonomy. JFR **27**(5), 534–560 (2010)
8. Goldberg, S.B., Maimone, M.W., Matthies, L.: Stereo vision and rover navigation software for planetary exploration. In IEEE Aerospace Conference Proceedings (2002)
9. Jackel, L.D., Krotkov, E., Perschbacher, M., Pippine, J., Sullivan, C.: The DARPA LAGR program: Goals, challenges, methodology, and phase I results. JFR **23**(11–12), 945–973 (2006)

10. Krajnk, T., Faigl, J., Vonsek, V., Konar, K., Kulich, M., Peuil, L.: Simple yet stable bearing-only navigation. JFR **27**(5), 511–533 (2010)
11. Krüsi, P., Bücheler, B., Pomerleau, F., Schwesinger, U., Siegwart, R., Furgale P.: Lighting-Invariant Adaptive Route Following Using ICP. JFR (2014)
12. Linegar, C., Churchill, W., Newman, P.: Work Smart. Recalling relevant experiences for vast-scale but time-constrained localisation. In: ICRA, Not Hard (2015)
13. MacTavish, K., Barfoot, T.D.: Towards hierarchical place recognition for long-term autonomy. In: ICRA Workshop (2014)
14. MacTavish, K., Paton, M., Barfoot, T.D.: Visual triage: a bag-of-words experience selector for long-term visual route following. In: ICRA (2017)
15. McManus, C., Furgale, P., Stenning, B., Barfoot, T.D.: Visual teach and repeat using appearance-based lidar. In: ICRA (2012)
16. Mhlfellner, P., Brki, M., Bosse, M., Derendarz, W., Philippsen, R., Furgale, P.: Summary maps for lifelong visual localization. JFR (2015)
17. Ostafew, C., Schoellig, A.P., Barfoot, T.D., Collier, J.: Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. JFR **33**(1), 133–152 (2016)
18. Paton, M., MacTavish, K., Ostafew, C., Pomerleau, F., Barfoot, T.D.: Expanding the limits of vision-based localization for long-term route following autonomy. JFR **34**(1), 98–122 (2017)
19. Paton, M., MacTavish, K., Warren, M., Barfoot, T.D.: Bridging the appearance gap: Multi-experience localization for long-term visual teach & repeat. In: IROS (2016)
20. Sibley, G., Mei, C., Reid, I., Newman, P.: Adaptive relative bundle adjustment. In: RSS (2009)
21. van Es, K., Barfoot, T.D.: Being in two places at once: Smooth visual path following on globally inconsistent pose graphs. In: CRV (2015)