

Learning Camera Performance Models for Active Multi-Camera Visual Teach and Repeat

Matias Mattamala, Milad Ramezani, Marco Camurri, and Maurice Fallon

Abstract—In dynamic and cramped industrial environments, achieving reliable Visual Teach and Repeat (VT&R) with a single-camera is challenging. In this work, we develop a robust method for non-synchronised multi-camera VT&R. Our contribution are expected Camera Performance Models (CPM) which evaluates the camera streams from the teach step to determine the most informative one for localization during the repeat step. By actively selecting the most suitable camera for localization, we are able to successfully complete missions when one of the cameras is occluded, faces into feature poor locations or if the environment has changed. Furthermore, we explore the specific challenges of achieving VT&R on a dynamic quadruped robot, ANYmal. The camera does not follow a linear path (due to the walking gait and holonomicity) such that precise path-following cannot be achieved. Our experiments feature forward and backward facing stereo cameras showing VT&R performance in cluttered indoor and outdoor scenarios. We compared the trajectories the robot executed during the repeat steps demonstrating typical tracking precision of less than 10 cm on average. With a view towards omni-directional localization, we show how the approach generalizes to four cameras in simulation.

I. INTRODUCTION

Following previously traversed paths is a useful capability for mobile robots. This is essential for missions, such as autonomous inspection and monitoring, where the same path is repeatedly traversed. This has motivated research into mapping and localization systems [1]. In particular, vision-based navigation systems such as Visual Teach and Repeat (VT&R) [2] have enabled different robots to repeat known routes without requiring metrically accurate maps. Visual sensors are inexpensive, lightweight, and provide both appearance and geometric information about the robot’s surroundings.

We are interested in legged robots, which are promising for inspection tasks due to their versatile mobility on challenging terrains. However, quadrupeds such as ANYmal [3] are holonomic and move with dynamic gaits, such as trotting and climbing stairs. These motions cause rapid feature change, blur and tracking failure, making it difficult to achieve VT&R with a single camera. Since cameras have a limited Field-of-View (FoV), redundancy in visual sensing, i.e using multiple cameras, allows such platforms to increase their vision capabilities, but at an increased computational cost and integration complexity (synchronization and calibration).

In this work, we present a visual navigation system for mobile robots based on the VT&R paradigm that takes advantage of multiple cameras to stay localized in presence

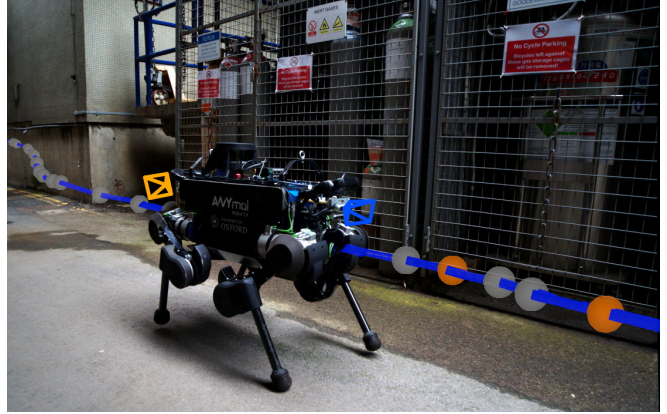


Fig. 1: Visual Teach and Repeat (VT&R) allowed us to quickly deploy the ANYmal robot for industrial routine inspections. In this work, we extend previous approaches by augmenting the topometric map with a *Camera Performance Model* (CPM). This allows us to dynamically choose the most reliable camera during a repeat, such as the front (blue) and rear (orange) cameras illustrated.

of clutter in narrow spaces (Fig. 1). In contrast to previous approaches, which typically process hardware synchronized cameras simultaneously [4], [5], [6], we instead select the camera providing the best performance for each segment of the path. This approach allows us to achieve accurate path tracking while also being robust to dynamic changes in the environment. Since cameras hardware synchronization is not required, our approach is more flexible, scalable and easier to deploy than traditional methods. The contributions of our work are summarized as follows:

- A VT&R system that uses multiple cameras during the *teach* step and learns performance models for each stream. These models are used in the *repeat* step to actively select the most informative camera.
- Deployment of our VT&R system on a quadruped robot, which enables it to autonomously follow a path it has traveled before using only vision, in spite of occlusions and dynamic locomotion gaits.
- Evaluation of the system in simulated and real scenarios with an ANYmal quadruped where peculiarities of legged systems are discussed. To the best of our knowledge, this is the first academic demonstration of VT&R on a legged platform.

The remainder of this paper is structured as follows. Sec. II discusses the related work. Sec. III describes our Active Multi-Camera VT&R system. Experimental results are presented in Sec. IV and conclusions are drawn in Sec. V.

II. RELATED WORK

This section discusses previous VT&R approaches as well as methods that exploit mapping or teach steps to improve the performance in subsequent traversals.

A. Visual Teach and Repeat

A variety of VT&R systems have been developed for wheeled robots [7], [8] and drones [9], [10], [11]. The main idea behind VT&R is that a topo-metric feature map is collected during a teach run. The map can then be used to guide the robot along the path learned during the teach run. Only local consistency between the path and the map is required to achieve path following.

In the past 10 years, most research on VT&R has been focused on improving the robustness against long-term environmental changes, which can compromise visual navigation. The problem has been commonly approached by creating and analyzing a varied set of traversals of the same route (also called *experiences*). This approach is known as *Experience-Based Navigation (EBN)* [12] or *Multi-Experience Localization (MEL)* [13]. Both systems have been deployed in autonomous cars, ground, and aerial platforms, with emphasis on seasonal and day-night reliability [14], [15], [10].

In the past, using multiple cameras in the context of VT&R has only been applied to deal with appearance changes. Paton *et al.* [16] used front-view and rear-view synchronized cameras on a Husky robot to make their VT&R system more robust to lighting conditions. However, this process was passive, as both cameras were processed together, and no prior information about the path was utilized.

In this work, we are less concerned about long term changes, such as day-night shifts or seasonal changes. Instead, we focus on abrupt changes, such as motion blur due to aggressive motions, occlusions caused by people or vehicles, camera exposure changes, and rapid scene change in cluttered locations. To this end, we also explore the use of multiple cameras, but we actively select the most informative camera during the repeat step. This is done by comparing online the current and the expected performance inferred from previous traversals.

B. Leveraging Past Experiences

Because VT&R systems have an explicit teach step, the knowledge collected in this step can be used to optimize the performance during the repeat execution. As with other methods, the main assumption is that the route taken does not change drastically, so the collected information (features and models) can be leveraged in future operation.

The work of Churchill *et al.* [17] collected localization statistics from several teach passes to train a Gaussian Process (GP) model of the *localization envelope* of a given path. This embedded the localization performance of the system, and it was used to predict potential failures.

Ondruska *et al.* [18] showed how to reduce the energy requirements of a planetary rover by scheduling when to use its cameras. They showed that significant energy savings could be achieved while still reliably localizing in open,

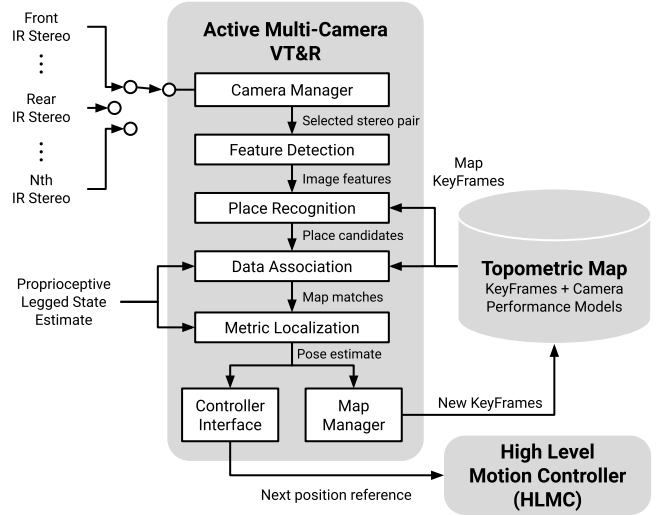


Fig. 2: Block diagram of our active multi-camera VT&R system. We augmented the topo-metric map with a Camera Performance Model (CPM) for each camera. The Camera Manager queries the models online to select the active camera.

park-like environments. Warren *et al.* [19] exploited previous VT&R experiences on a drone to actively control a gimbal system so as to reduce the orientation error between the camera and the viewpoint used while recording experiences.

Recently, Zhang *et al.* presented a perception-aware navigation approach, named *Fisher Information Fields* [20], [21]. This map representation allocates the expected localization performance (given by the Fisher Information matrix) in a discrete grid, which is used to compute the expected information for arbitrary poses. While we base our localization performance metrics on similar principles, in this work we focus on topo-metric representations instead.

III. METHOD

Our goal is to develop a multi-camera Visual Teach and Repeat system for robots with multiple cameras, with a focus on quadruped robots.

A. System Overview

The main modules of our system are shown in Fig. 2. The structure follows the approach taken by Furgale and Barfoot [2], with three main differences:

- We use a slowly drifting proprioceptive state estimate (for a legged robot, provided by a system such as TSIF [22]) instead of visual odometry to simplify the mapping process and reduce the computational burden in the teach step (Sec. III-B).
- The topo-metric map is augmented with a *Camera Performance Model (CPM)* for each camera, which is learned using the teach trajectory (Sec. III-C).
- A new *Camera Manager* module determines which camera should be processed at the current instance. During the teach, the manager processes every camera in turn. During repeat, it exploits the CPM to select the most suitable camera for a specific part of the path (Sec. III-D).

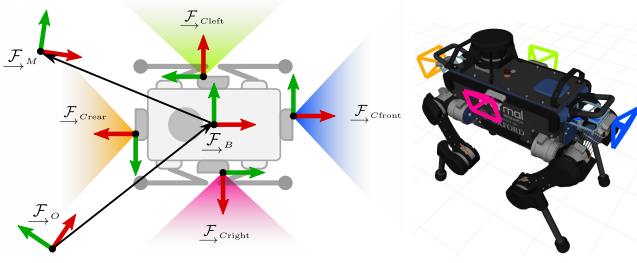


Fig. 3: Top-view diagram of the frames and color convention to identify the cameras throughout this paper.

We consider that up to 4 cameras could be attached to the main body (front, rear, left and right). Fig. 3 illustrates the coordinate frames and the position of the cameras.

The proprioceptive state estimate is defined in the fixed odometry frame \mathcal{F}_O , while the VT&R localization is defined in the fixed map frame \mathcal{F}_M corresponding to the teach path. The moving frame \mathcal{F}_B is rigidly attached to the robot's chassis, as well as the four camera frames \mathcal{F}_C .

B. Topo-metric Mapping with Multiple Cameras

In our system, the teach step builds a topo-metric map of the path over which the robot is teleoperated. The map is represented by a collection of *keyframes* \mathcal{K} connected by relative transformations (as in [23]) which are expressed in the map frame \mathcal{F}_M relative to the body frame \mathcal{F}_B . Each keyframe $k \in \mathcal{K}$ stores:

- A stereo image pair, tagged with the source camera.
- Triangulated AKAZE [24] features \mathcal{P} for metric pose estimation relative to the keyframe.
- A Bag-of-Visual-Words vector, based on DBoW2 [25].
- The body pose of the keyframe in the map frame ${}_M\mathbf{T}_{MB} \in \text{SE}(3)$.
- The intrinsic calibration between the camera and the body pose ${}_B\mathbf{T}_{BC} \in \text{SE}(3)$.
- A CPM of each available camera in the current setup.

The teach step performs the mapping process using a single camera at a time. For each stereo image frame, the *Feature Detector* module extracts features \mathcal{Z} , which are then matched against the current map by the *Data Association* module. We use the quadruped's proprioceptive state estimate as a motion prior which helps to guide feature matching.

The *Metric Localization* module uses the matches to perform a registration against the map points \mathcal{P} in the last created keyframe: we first use Perspective-n-Points (PnP) to obtain an initial estimate, which is later refined via pose-only optimization using the reprojection residual $\mathbf{r}_{\text{reproj}}(\mathbf{z}, \mathbf{p})$ with covariance Σ_{reproj} , and a pose prior residual $\mathbf{T}_{\text{prior}}$ with covariance Σ_{prior} :

$$\arg \min \sum_{\mathbf{z} \in \mathcal{Z}, \mathbf{p} \in \mathcal{P}} \|\mathbf{r}_{\text{reproj}}(\mathbf{z}, \mathbf{p})\|_+^2 + \|\mathbf{r}_{\text{prior}}(\mathbf{T}_{\text{prior}})\|_{\Sigma_{\text{prior}}}^2 \quad (1)$$

The main goal of this optimization is to recover an estimate of the covariance Σ_{visual} corresponding to the visual pose estimate $\mathbf{T}_{\text{visual}}$. Further, the covariance is used to compute the negative entropy E as follows:

$$E = -\log(|\Sigma_{\text{visual}}|) \quad (2)$$

E is single scalar that characterizes the performance of the localization at a given pose: a larger negative entropy implies a better localization estimate, and vice-versa. An advantage of this method over other criteria, such as the number of tracked features, is that it characterizes the whole localization process. For instance, tracking a small number of nearby features or a large number of distant features are treated similarly, since both situations can lead to poor localization estimates. E is of particular importance for our system because:

- It is used by the *Map Manager* module as a criterion when creating new keyframes, using the running average filter strategy proposed by Kuo *et al.* [26].
- The poses and negative entropies of frames that are not used to create new keyframes are stored in the map as *performance samples* \mathcal{S} of the actual camera in a specific part of the path.

The process is executed for each camera individually, so as to sample their performance assuming no other cameras are available. This could generate inconsistent trajectories for each camera, so we enforce smoothness along the path by prioritizing the use of the proprioceptive state estimate for the teach step. The output is shown in Fig. 4 (a).

C. Learning Performance Models for Each Camera

As previously described, the teach step generates not only a topo-metric map, but also a set of performance samples \mathcal{S} from the whole path. A performance sample s is defined as a tuple (\mathbf{T}_s, E_s, c_s) , where \mathbf{T}_s is the pose of the sample expressed in the map frame, E_s is the negative entropy computed for that specific pose, and c_s is a tag to identify the particular camera that generated that sample.

We use the samples to learn a CPM, which is a model that embeds the performance of a camera for a given teach path. The CPM for a single camera c is expressed by a collection of Gaussian distributions defined for every keyframe in the path. Their parameters (μ_c, σ_c) are the result of the learning process. We define CPMs for all the cameras available in the robot, and we are able to query them at each keyframe of the path to determine which camera is likely to provide the best localization estimate.

The learning algorithm, defined in Alg. 1, performs a spatially weighted averaging of samples around each keyframe. The averaging weights are computed using a *radial basis function* kernel [27] denoted by $\kappa(\mathbf{T}_1, \mathbf{T}_2)$ and defined as:

$$\kappa(\mathbf{T}_1, \mathbf{T}_2) = \exp\left(-\frac{d(\mathbf{T}_1, \mathbf{T}_2)^2}{2l}\right) \quad (3)$$

where, $d(\cdot, \cdot)$ is a function that computes the distance between the two poses, ignoring the rotational component, while l is a hyperparameter (scale length) which controls smoothness. Fig. 4 (b) shows the output of the learning process for all the keyframes along a given path. This is similar to Gaussian Process regression, but is defined on the discrete space of keyframes, independently from spatial coordinates.

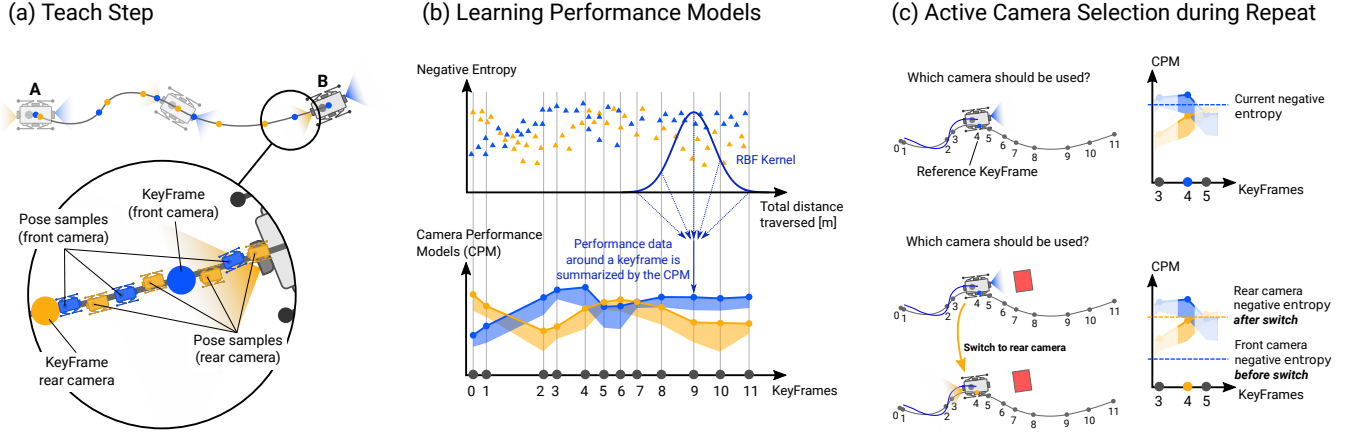


Fig. 4: Main steps of our VT&R system. **(a) Teach step:** The robot is first teleoperated from A to B to build a topo-metric map of the environment. Keyframes are created for each camera, denoted by the different colors. Along with the map, sampled poses and negative entropies are also computed and stored. **(b) Learning Performance Models:** For each keyframe, the closest negative entropy samples within a radius d_{\max} are grouped and are averaged using an RBF kernel to learn a model of performance for each camera (Alg. 1). **(c) Repeat step with active camera selection:** The learned CPMs are used to select the camera with highest predicted performance at each segment of the map, or to change the selected camera if the predicted performance is not as expected.

Algorithm 1: CPM learning using teach path

Input: Keyframes \mathcal{K} , performance samples \mathcal{S} ,
maximum distance for sample search d_{\max} ,
kernel weighting function $\kappa(\cdot, \cdot)$

Output: CPM for each camera

```

foreach keyframe  $k$  in  $\mathcal{K}$  do
   $\mathbf{T}_k \leftarrow \text{GetKeyFramePose}(k)$ 
   $\mathcal{S}^* \leftarrow \text{SearchSamplesWithinRadius}(k, \mathcal{S}, d_{\max})$ 
  foreach camera  $c$  in  $\mathcal{S}^*$  do
     $w_c \leftarrow \sum_{s \in \mathcal{S}^*} \kappa(\mathbf{T}_s, \mathbf{T}_k)$ 
     $\mu_c \leftarrow 1/w_c \sum_{s \in \mathcal{S}^*} E_c \kappa(\mathbf{T}_s, \mathbf{T}_k)$ 
     $\sigma_c \leftarrow \sqrt{1/w_c \sum_{s \in \mathcal{S}^*} (E_c - \mu_c)^2 \kappa(\mathbf{T}_s, \mathbf{T}_k)}$ 
    UpdateCPM( $k, c, \mu_c, \sigma_c$ )

```

D. Repeat Step with Active Camera Selection

The repeat step involves different procedures depending on the status of the system:

a) Global Relocalization: If the system is initializing, the status is *lost*, and an arbitrary camera will be chosen to attempt relocalization. First, the Place Recognition module searches candidate keyframes using Bag-of-Words. Then, the Data Association module performs a standard descriptor matching. The matches are later verified by the Metric Localization module by attempting a PnP registration and then using optimization refinement to discard outliers. The keyframe with the most matches is selected as the reference keyframe and the system status is set to *localized*. Since the previous procedure is agnostic to which camera generated the keyframe, we also compute the reference keyframes for

the other available cameras. This is done by checking all the neighbor keyframes and associating to each camera the closest keyframe with the most similar orientation.

b) Path Traversal with Active Camera Selection: Once the system has computed the reference keyframes and an initial pose has been estimated, the repeat step is ready for execution. Details regarding the integration with the quadruped's controller are described later in Sec. III-E.

While the robot traverses the path and the Feature Detection, Data Association, and Metric Localization are being executed for the teach step, the Camera Manager module analyzes the different image streams and *actively* changes the current camera if: 1) there is another camera that can provide better performance at that specific part of the path, or 2) the current performance is not as the model describes, typically due to a change in the environment.

The first case is straightforward: it only needs to query each μ_c in the CPM to find the best camera for the current reference keyframe. The second case requires comparison between the current negative entropy E_t and a lower bound $E_t < \mu_c - k\sigma_c$ computed from the CPM parameters (μ_c, σ_c) associated to the reference keyframe. The bound defines a *margin* so as to not select a new camera unless performance has decreased significantly, which can be tuned with the hyperparameter k .

In general, with accurate path tracking and with similar visual conditions in the teach and repeat steps, the previous inequality will never be satisfied, and the negative entropy will stay within the limits. However, if the environment changes, the feature extraction will be affected, potentially degrading the visual pose estimate and decreasing the negative entropy below the lower bound. When this occurs, the camera is flagged and cannot be used for a fixed time. The remaining CPMs are queried to find the next best camera for the given path section. An example of this procedure is illustrated in Fig. 4 (c). If this procedure fails for all the

cameras and the system loses visual tracking, the system reports *Tracking Lost*.

c) *Tracking Lost*: When the system loses visual tracking, we use the last successful localization estimate and predict the current pose by using relative motion estimates from the legged proprioceptive state estimator. Meanwhile, the system will attempt a re-localization by matching against the neighbor keyframes within a certain radius, a procedure we call *Local Relocalization*. If the system cannot succeed after 10 seconds, it declares itself *lost* and will stop navigating until it is reset by the user.

E. Closed-loop Integration in a Legged Robot

The motion of a wheeled robot or a car is smooth and without any sharp jerks. A state-of-the-art VT&R system can track precisely enough to keep a UGV within the tram-lines of previous runs (as in [2]). This degree of smoothness has a stabilizing effect on VT&R localization. In contrast, the same degree of smoothness in camera motion is not possible on legged robots. The robot's gait induces a sharp, jerky motion, so that exact teach trajectory tracking is impossible.

The quadruped's Whole Body Controller (WBC) controls its 12 joints to achieve goals such as a desired base velocity. We interface with it through a High Level Motion Controller (HLMC), which computes a base velocity reference given a desired base pose. The VT&R system generates a sequence of waypoints from the teach path expressed in the map frame. Given the current localization estimate, the VT&R system selects the closest waypoint to the robot and sends it as the next desired base pose reference to the HLMC. In this way, we circumvent the need to precisely replicate the base motions of original teach trajectory, but we keep the robot close to it at all times.

Finally, in contrast to wheeled platforms, legged robots are holonomic and can strafe or turn in place to execute inspection tasks; we illustrate how the VT&R handles such situations in our attached video.

IV. EXPERIMENTAL RESULTS

Experiments were performed with an ANYmal B300 robot equipped with two unsynchronized Intel RealSense D435i stereo cameras angled down by 12° ; we used the IR stereo pairs as the visual input for our system. The robot also carries a Velodyne VLP-16 LiDAR, which we used in post-processing to obtain 10 Hz ground truth by registering scans within a prior map using Iterative Closest Point (ICP) [28]. Our system ran onboard on a single Intel i5 CPU shared along with other required modules and drivers.

For evaluation, we compare repeat trajectories to the initial teach run. We determined the instantaneous tracking error as the perpendicular distance between each pose during the repeat run and a line fit to the nearest neighbor points of the teach step path. The mean *Path Tracking Error* (PTE) is a measure of tracking performance for a full run.

Our VT&R system was tested in two experimental scenarios: an indoor workshop (E1) and a larger industrial environment outdoor (E2). Tab. I summarizes the path tracking

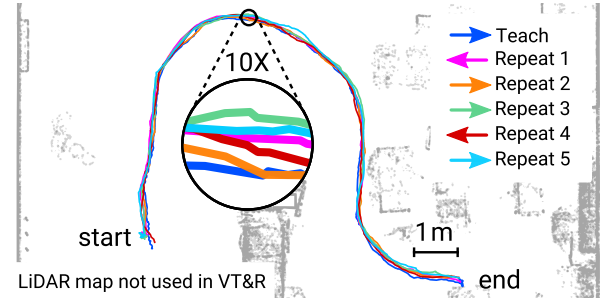


Fig. 5: Experiment 1 (Indoor): Ground truth trajectories of teach (blue) and repeat runs. Direction of motion is indicated by the legend. Overall, the robot never exceeded 20 cm of path tracking error, with an average of 7 cm.

Path Tracking Error (PTE) $\mu \pm \sigma$ [m]					
	R1	R2	R3	R4	R5
E1	0.05 ± 0.03	0.06 ± 0.03	0.09 ± 0.06	0.06 ± 0.03	0.09 ± 0.04
E2	0.09 ± 0.04	0.13 ± 0.05	0.07 ± 0.05	0.14 ± 0.07	-

TABLE I: Quantitative results for the indoor (E1, 5 repeats) and outdoor experiments (E2, 4 repeats).

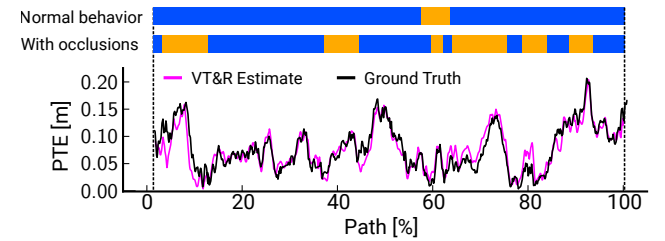


Fig. 6: Indoor experiment (E1): Estimated (magenta) and ground truth (black) PTE between teach and repeat R5. Color bars show the switch between the front camera (blue) and the rear camera (orange) in normal operation (top) and when occluded (bottom).

performance for all the runs using the VT&R pose estimates. Additionally, we ran our system in simulation with 4 cameras to demonstrate the advantages of our approach with more complex camera setups.

A. Experiment 1: Indoor

We first performed a series of experiments in a cluttered lab environment. The robot was teleoperated to walk between furniture, machines and other equipment, covering a distance of 15 m. It then autonomously returned to the initial position (backwards), and repeated the path back and forth 5 times.

The robot demonstrated stable navigation in all these runs, and it was able to stay within 20 cm of the teach path at all times, regardless of the walking direction (Fig. 5). Numerical comparisons in Tab. I, demonstrate the low tracking error obtained for each run.

In Fig. 6, we evaluated the tracking performance (as estimated by VT&R system online) by comparing it to the true tracking error (computed using the LiDAR ground truth) for the fifth repeat run of these experiments. The high degrees of correlation between the two estimates demonstrates that the VT&R system can accurately localize the robot against

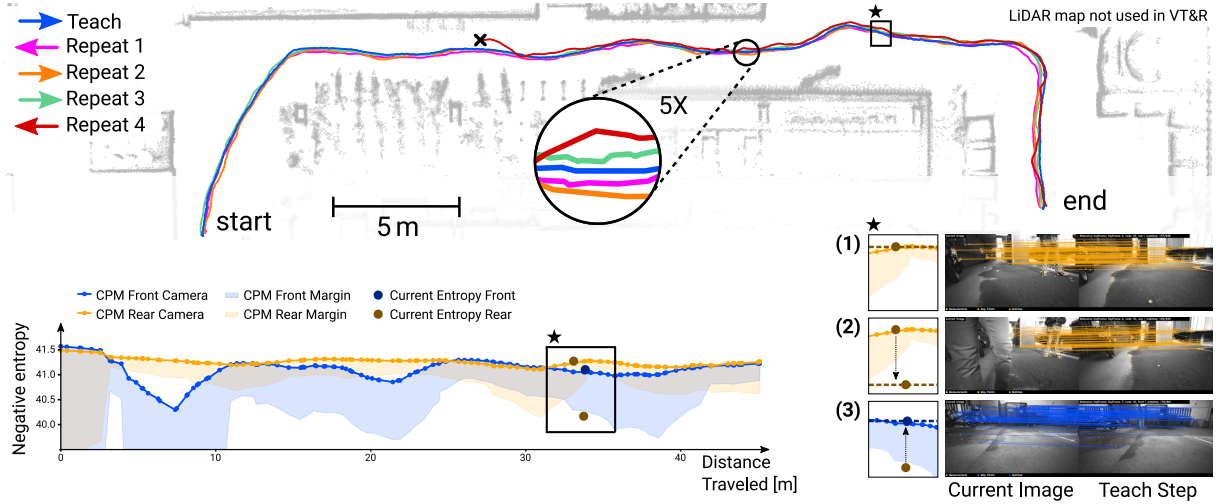


Fig. 7: Outdoor experiment (E2): *Top*: Ground truth trajectories of the teach (blue) and repeat runs. The robot never exceeded 20 cm of tracking error, with an average of 11 cm. *Bottom*: Left plot illustrates the CPM computed for the whole path, right images are examples of matches between the live stream and the teach path. A segment of the trajectory during Repeat 2 in which the camera is occluded is marked with a ★ symbol: (1) Rear camera performance (in orange) is within the CPM limits. (2) The camera is occluded, leading to a drop in the negative entropy, triggering a camera switch. (3) After the system switched to the front camera, its current negative entropy (blue) is closer to the CPM prediction.

the teach path even if one of the cameras is occluded. Deviations are due to the shape of the teach path and the responsiveness of the tracking controller.

B. Experiment 2: Outdoor

For our second experiment, we tested our system in an outdoor environment with adverse lighting and repetitive, industrial structure. We teleoperated the robot to walk a 45 meter long path, which was successfully traversed 3 times in repeat mode (Fig. 7). On a fourth repeat run the system was interrupted after the localization module diverged due to poor visual feature tracking. It was caused by changing lighting conditions, which is subject to future work.

During the second repeat run we occluded the cameras by having a person walk in front of the robot. Our VT&R successfully changed the active camera and completed the mission regardless.

C. Experiment 3: Qualitative Experiments with 4 Cameras

Lastly, we performed experiments in simulation by equipping the ANYmal with 4 cameras. The goal was to demonstrate that our approach naturally generalizes to other camera configurations and is applicable to the latest version of ANYmal, the C-series, which has a similar 4 camera configuration. For the teach step, we made the robot walk through the environment with motion in all directions (forward, sideways and turning). Fig. 8 shows an example of the trajectories traversed in the simulation. Further experiments with the real robot will be a focus of future work.

V. CONCLUSIONS

We presented a novel VT&R system that utilizes multiple non-synchronized cameras to perform autonomous point-to-point navigation. By exploiting information collected during

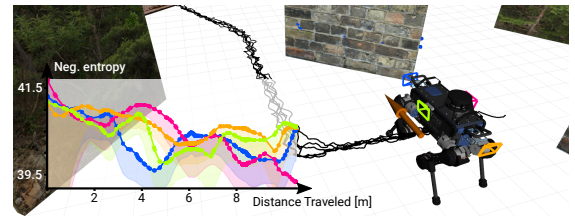


Fig. 8: Experiment 3 (Simulation): Tracking performance and CPM with 4 cameras. Black lines denote the ground truth paths for 6 consecutive repeats. The orange arrow is the next waypoint sent to the HLMC.

a teach step, we learned a performance model for each camera that preserved the topo-metric structure. We demonstrated how the system utilized the learned models online to actively select the most informative camera and be resilient to sudden changes in the environment due to occlusions.

In a series of real and simulated navigation scenarios on a quadruped robot, our system successfully followed a previously taught route in spite of the complexities of jerky motion and people (intentionally) occluding the cameras.

In future, we plan to extend our VT&R system with other visual cues to improve its performance in more complex locomotion regimes such as stair climbing and obstacle traversals which cause the visual scene to change more dramatically.

ACKNOWLEDGEMENT

This research is supported by the ESPRC/UKRI ORCA Robotics Hub (EP/R026173/1), a Royal Society University Research Fellowship (Fallon) and the National Agency for Research and Development of Chile (ANID) / Scholarship Program / DOCTORADO BECAS CHILE / 2019-72200291 (Mattamala).

REFERENCES

- [1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [2] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of Field Robotics*, vol. 27, no. 5, pp. 534–560, 2010.
- [3] M. Hutter, C. Gehring, A. Lauber, F. Gunther, C. D. Bellicoso, V. Tsounis, P. Fankhauser, R. Diethelm, S. Bachmann, M. Bloesch, H. Kolvenbach, M. Bjelonic, L. Isler, and K. Meyer, "ANYmal - toward legged robots for harsh environments," *Advanced Robotics*, vol. 31, no. 17, pp. 918–931, 2017.
- [4] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.
- [5] P. Liu, M. Geppert, L. Heng, T. Sattler, A. Geiger, and M. Pollefeys, "Towards Robust Visual Odometry with a Multi-Camera System," *IEEE International Conference on Intelligent Robots and Systems*, pp. 1154–1161, 2018.
- [6] C. Won, H. Seok, Z. Cui, M. Pollefeys, and J. Lim, "OmniSLAM: Omnidirectional Localization and Dense Mapping for Wide-baseline Multi-camera Systems," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.
- [7] P. Furgale and T. Barfoot, "Stereo mapping and localization for long-range path following on rough terrain," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2010, pp. 4410–4416.
- [8] T. Krajník, F. Majer, L. Halodova, and T. Vintr, "Navigation without localisation: Reliable teach and repeat based on the convergence theorem," in *IEEE International Conference on Intelligent Robots and Systems*, 2018, pp. 1657–1664.
- [9] F. Gao, L. Wang, B. Zhou, X. Zhou, J. Pan, and S. Shen, "Teach-Repeat-Replan: A Complete and Robust System for Aggressive Flight in Complex Environments," *IEEE Transactions on Robotics*, pp. 1–20, 2020.
- [10] M. Warren, M. Greeff, B. Patel, J. Collier, A. P. Schoellig, and T. D. Barfoot, "There's no place like home: Visual teach and repeat for emergency return of multirotor UAVs during GPS failure," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 161–168, 2019.
- [11] M. Nitsche, F. Pessacg, and J. Civera, "Visual-inertial teach and repeat," *Robotics and Autonomous Systems*, vol. 131, p. 103577, 2020.
- [12] W. Churchill and P. Newman, "Practice makes perfect? Managing and leveraging visual experiences for lifelong navigation," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4525–4532, 2012.
- [13] M. Paton, K. Mactavish, M. Warren, and T. D. Barfoot, "Bridging the appearance gap: Multi-experience localization for long-term visual teach and repeat," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, pp. 1918–1925, 2016.
- [14] C. Linegar, W. Churchill, and P. Newman, "Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, pp. 787–794, 2016.
- [15] M. Paton, F. Pomerleau, K. MacTavish, C. J. Ostafew, and T. D. Barfoot, "Expanding the Limits of Vision-based Localization for Long-term Route-following Autonomy," *Journal of Field Robotics*, vol. 34, no. 1, pp. 98–122, 2017.
- [16] M. Paton, F. Pomerleau, and T. D. Barfoot, "Eyes in the Back of Your Head: Robust Visual Teach & Repeat Using Multiple Stereo Cameras," *Proceedings - 2015 12th Conference on Computer and Robot Vision, CRV 2015*, no. June, pp. 46–53, 2015.
- [17] W. Churchill, C. H. Tong, C. Gurău, I. Posner, and P. Newman, "Know your limits: Embedding localiser performance models in teach and repeat maps," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 4238–4244, 2015.
- [18] P. Ondruška, C. Gurău, L. Marchegiani, C. H. Tong, and I. Posner, "Scheduled perception for energy-efficient path following," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 4799–4806, 2015.
- [19] M. Warren, A. P. Schoellig, and T. D. Barfoot, "Level-Headed: Evaluating Gimbal-Stabilised Visual Teach and Repeat for Improved Localisation Performance," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 7239–7246, 2018.
- [20] Z. Zhang and D. Scaramuzza, "Perception-aware receding horizon navigation for MAVs," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2534–2541, 2018.
- [21] Z. Zhang and D. Scaramuzza, "Beyond point clouds: Fisher information field for active visual localization," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 5986–5992.
- [22] M. Bloesch, M. Burri, H. Sommer, R. Siegwart, and M. Hutter, "The Two-State Implicit Filter Recursive Estimation for Mobile Robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 573–580, 2018.
- [23] W. Churchill and P. Newman, "Experience Based Navigation: Theory, Practice and Implementation," Ph.D. dissertation, University of Oxford, 2012. [Online]. Available: <http://www.robots.ox.ac.uk/~mobile/Theses/WinstonThesis.pdf>
- [24] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *BMVC 2013 - Electronic Proceedings of the British Machine Vision Conference 2013*, 2013.
- [25] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, oct 2012.
- [26] J. Kuo, M. Muglikar, Z. Zhang, and D. Scaramuzza, "Redesigning SLAM for Arbitrary Multi-Camera Systems," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.
- [27] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Science+Business Media Inc., 2006.
- [28] P. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, feb 1992.