2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

TRADE: Object Tracking with 3D Trajectory and Ground Depth Estimates for UAVs

Pedro F. Proença*, Patrick Spieler*, Robert A. Hewitt*, Jeff Delaune*

***Abstract*— We propose TRADE for robust tracking and 3D localization of a moving target in complex environments, from UAVs equipped with a single camera. Ultimately TRADE enables 3d-aware target following.**

Tracking-by-detection approaches are vulnerable to target switching, especially between similar objects. Thus, TRADE predicts and incorporates the target 3D trajectory to select the right target from the tracker's response map. Unlike static environments, depth estimation of a moving target from a single camera is an ill-posed problem. Therefore we propose a novel 3D localization method for ground targets on complex terrain. It reasons about scene geometry by combining ground plane segmentation, depth-from-motion and single-image depth estimation. The benefits of using TRADE are demonstrated as tracking robustness and depth accuracy on several dynamic scenes simulated in this work. Additionally, we demonstrate autonomous target following using a thermal camera by running TRADE on a quadcopter's board computer.

I. Introduction

Object tracking and 3D localization from a UAV has several applications (e.g. defense, disaster response, wildlife monitoring) involving target following, which already have some commercial solutions for consumer drones (e.g. Skydio, DJI) relying on GPS beacons, stereo cameras and visual object trackers. However persistent tracking and 3D localization of a non-cooperative target from a single camera remains a challenging problem.

In terms of persistent tracking, *Tracking-by-detection* has become the dominant paradigm [1, 2] in generic visual object tracking thanks to learned discriminative and efficient models. Despite its success, this approach alone leads to *target switching* especially between similar objects (as shown in Fig. 1). This is mainly due to the absence of an object motion model. In this work, we propose to predict the 3D trajectory of a ground object with visual-inertial odometry (VIO) to prevent *target switching* by selecting the peak from the tracker's correlation filter response map closer to the predicted location.

Unlike static environments, estimating the depth of a moving object from a single camera is a ill-posed problem without knowledge of the object's motion. Thus we propose a solution for ground targets by combining single-image depth estimation with depth estimates from camera motion. While the former is used to obtain dense depth of the moving target relative to its surroundings, the latter provides sparse accurate depth measurements from the terrain. Our localization method can then infer the ground plane from the scene geometry (i.e. ground plane segmentation) to raycast the object's depth. Moreover, a temporal plane fusion step is proposed to account for temporally covered or textureless ground and missing depth-from-motion due to hovering.

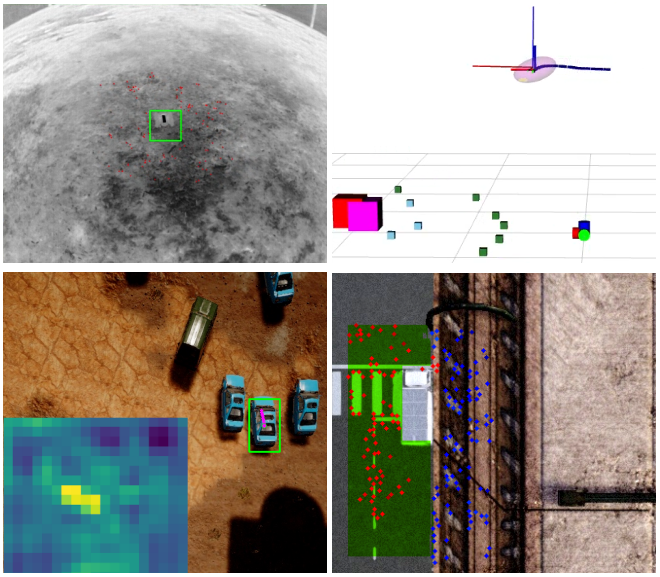

Fig. 1: *Top*: Drone following autonomously a remote-controlled car using a thermal camera with onboard TRADE https://youtu.be/LK3tCvKXF6w. *Bottom:* Two scenes from our synthetic dataset showing the benefit of our approach. *Bottom-Left*: The tracker's correlation filter response [1] shows two similar peaks, which lead to *target switching* between cars. Our approach uses the trajectory predictions (seen in magenta) to select the right peak. *Bottom-Right:* UAV tracking a truck from a rooftop. Our ground plane segmentation (shown in green) allows us to reject features from the building top to estimate the correct ground plane. Tracked features are color-coded based on depth. For more details refer to: https://youtu.be/MGPK65gm9GI

Our contributions are the following:

- A novel 3D localization method for a dynamic ground object that is robust to high terrain relief.
- Couple object 3D trajectory forecasting, and camera pose with a Discriminative Correlation Filter tracker to avoid target switching.
- A photorealistic tracking dataset for UAV with ground-truth depth and poses and an extensive evaluation.
- Demonstrate real-time UAV target following using TRADE onboard.

* Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA

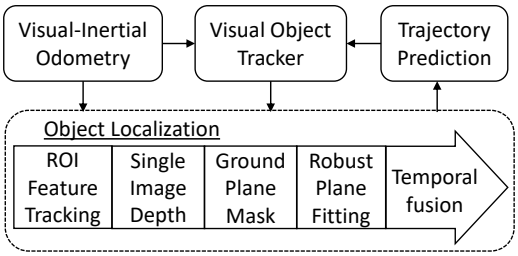

Fig. 2: Overview of our object tracking and 3D localization system.

II. Related Work

Recently, in generic visual object tracking, several learning-based efforts have been made to address the problem of *target switching* [3, 4]. In Multi-Object Tracking [5, 6], this problem is formulated as a data association problem which is commonly addressed by using a constant-velocity Kalman Filter and a Hungarian algorithm. However this Kalman Filter works on the image space, where optical flow is non-linear and it assumes a static camera. A camera motion compensation is used in [6] based on image registration and in [7] based on homography warping, which assumes the homography is estimated using the object's ground plane. In [8], target 3D trajectory is modeled in a SLAM factor graph but it relies on a stereo-camera. In [9, 10] trajectory models are learned for human motion using LSTMs.

Object 3D localization from a UAV has been addressed using GPS receivers [11], laser range finder [12], georeferenced topographic maps [13] and flat earth assumption [14]. There has been extensive work [15–17] using ground plane estimation for 3D object localization. The most similar to our work is [15], which uses depth estimates from Visual Odometry and a barometer to estimate the plane normals and height but this also assumes the scene is planar. In terms of monocular object 3D localization from the ground, [18] proposes to estimate 3D car poses by combining 2D bounding boxes, orientation regression and the object dimensions. Single-image depth networks [19, 20] have been demonstrating compelling results on several datasets (e.g KITTI). In this work, we investigate how these generalize to aerial downward-looking cameras.

III. System Overview

Our system pipeline is shown in Fig. 2. Firstly, a Discriminative Correlation Filter (DCF) tracker is initialized as usual with a bounding box on an initial frame. The bounding box is then used to initialize the ROI Feature Tracking by detecting Shi-Tomasi corners within a Region of Interest (ROI) surrounding the bounding box, which is excluded from the ROI. The ROI is then shifted as the bounding box moves through tracking. Using this dedicated feature tracking module allows to maintain a dense distribution of features around the object, without adding overhead to the VIO.

For every frame, depth is estimated and refined for the ROI tracks given the camera poses from VIO. These tracks can be backprojected to a point cloud and fit a plane. However, since not all tracks are from the object's ground plane, we first select tracks based on our ground plane segmentation which relies on a single image depth model to provide dense depth for both the target and the ROI. However since this is only relative depth, we use the ROI feature depth estimates to effectively scale it.

Given the resulting ground plane mask, the selected ROI tracks with 3D coordinates are used in a RANSAC multi-plane fitting routine. Since the ground plane segmentation can fail and the ROI features may not be enough, we use a temporally-fused plane model, which aggregates the inlier points from the last RANSAC plane fitting in a buffer together with inliers from past frames. The temporal fusion also includes a gating strategy to enforce temporal consistency. The aggregated points are used in a RANSAC multi-plane fitting loop once again to estimate the final plane. Then, given the target image coordinates from the DCF tracker and the camera pose we can raycast the 3D location. This is then used to update the trajectory model, whose predictions for the next frames are used to guide the DCF Tracker, as described in the next section. The remaining sections provide more details for each module of our pipeline.

IV. Tracking with Trajectory Estimates

Visual object trackers generally output a 2D score map (shown in Fig. 1) that maps to locations in an image search window around the previous target location. Then the location with highest score is simply selected as the new target location.

Instead, we first center the search window around the location predicted by our trajectory model which is projected to the current image using the camera pose. We then perform peak selection on the score map: First we normalize the score map with a softmax function, then, using Non-Maximum suppression, we select as location candidates the peaks within a certain fraction of the maximum peak and take the peak that is closer to the search window origin.

As a trajectory model, We use a linear Kalman filter to estimate the state $\{p, v, a\}$, respectively the object absolute 3D location, velocity and acceleration. To prevent unbounded motion during temporary tracking loss, we use a damping factor both in velocity and acceleration instead of a constant model in the state transition. The state is updated using only the 3D location observation residuals. The process and measurement noise was set empirically.

V. Robust Object 3D Localization

Our object localization is based on the projection of the object bounding box center on the ground plane. However as illustrated in Fig. 3.a, camera off-nadir β leads to a lateral error $\tilde{x} = h \tan \beta$ where h is the height at which the ray intersects the object. To reduce this error we can lift the ground plane by a prior estimate of half the object height before raycasting the depth. The next subsections cover all modules of our localization approach.

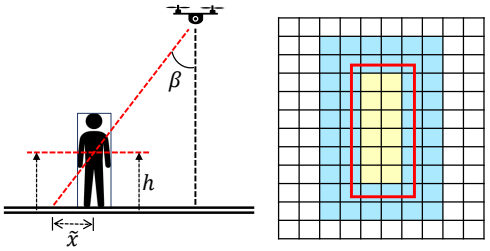

Fig. 3: (a) Lateral error due to off-nadir. To minimize this error we lift the ground plane before raycasting. (b) Image grid used for segmenting the ground plane from the cropped Single-Image depth map. The red square corresponds to the target bounding box. Points within the yellow tiles are used to estimate the object relative height. Planes are fit to blue and white tiles. Based on the estimated planes, seeds are sampled from the blue section for Region Growing, which is performed on the white tiles.

A. ROI Feature tracking

To track a dense distribution of static features around the target, we detect Harris corners from a tiled ROI surrounding the bounding box. For simplicity, The ROI is split by a 3×3 grid where the center tile corresponds to the object bounding box. For every tile (except the middle one), we maintain N tracks by adding more features if necessary. These are tracked frame-to-frame using KLT. For each track, we keep the first observation and respective camera pose and triangulate depth with the current camera pose using a linear solution. These stereo depth estimates are then fused using well-established recursive multiplication of Gaussians [21] and initialized as 3D points once their uncertainties drop below a certain threshold.

B. Single-Image Depth

We use MiDaS [19] with the authors supplied weights as our single-image depth model because it was trained across a large mixture of different datasets and it performed the best in our preliminary results. However MiDaS only generates an affine transformation of the world-scaled inverse depth . Therefore, we estimate the translation and scale parameters $\{\theta_0, \theta_1\}$ by minimizing

$$\{\theta_0, \theta_1\}^* = \underset{\{\theta_0,\theta_1\}}{\arg\min} \sum_{i}^{M} \left(\theta_0 + \rho_i' \theta_1 - \rho_i\right)^2 \qquad (1)$$

in a linear least squares where $\{\rho_1, ..., \rho_M\}$ are ROI feature inverse depth values and $\{\rho_1', ..., \rho_M'\}$ are the corresponding MiDaS map pixel values. Since the ROI depth values and MiDas may contain outliers we also minimize this in a RANSAC with a minimum set of 2 point correspondences. It is worth noting that the solution is degenerative on planes parallel to the image plane. Thus in the RANSAC sampling we enforce a minimum distance between ROI depth samples.

One might be tempted to sample directly depth on the target from the recovered world-scaled depth map. However the error is unbounded inside the bounding box, as shown in Fig. 4, as this does not contribute to the optimization.

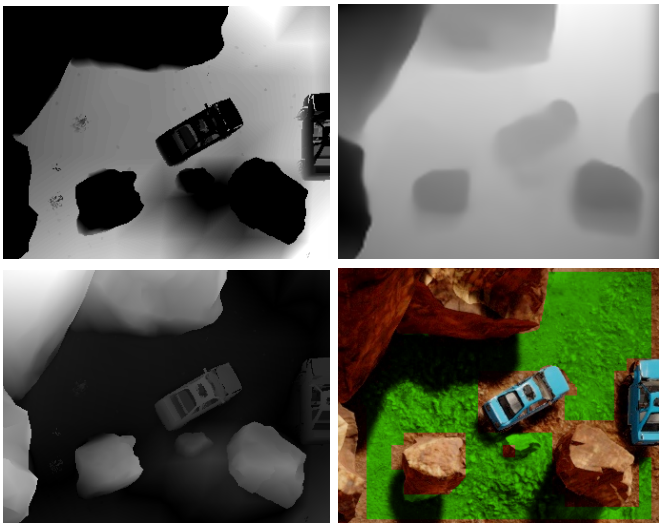

Fig. 4: *Top-left:* Ground-truth depth normalized with the Single-Image depth shown on *top-right*, after affine transformation. *Bottom-left:* Single-Image depth error. *Bottom-right:* Obtained ground mask after refinement shown in green.

C. Ground Plane Segmentation

To segment the ground plane we exploit the fact that the ground plane is below the object and that it is visible on the image near the object bounding box. To do so, first we crop the recovered Single-Image depth map – to fit the ROI active tracks, backproject it to an organized point cloud and rotate it using the camera pose to align it with gravity. We then estimate the object coarse height by averaging the height values of all points within the target bounding box.

The gravity-aligned point cloud is then segmented using the grid illustrated in Fig. 3.b. For each tile that is not contained in the target bounding box we fit plane to the respective points. Tiles that cross the bounding box are dilated outward forming a *seeding region* where we sample tiles for region growing. Specifically, first we remove tiles from the *seeding region* that have an average point height higher than the object's height estimate. We then select as a seed the tile with minimum Mean Squared Error (MSE) given the plane fitting. Tile-wise Region Growing is performed on the white tiles shown in Fig. 3.b, and alternated with the seeding to cover possible split ground plane as shown in Fig. 4. The Region Growing algorithm is inspired by [22], it uses a 4-neighbour search and proceeds as follows: A tile neighbour c of a current seed s is added to the growing region if: (i) the dot product of their plane normals exceeds a threshold, (ii) the distance from the centroid in c to the plane in s is lower than $d \sin \alpha$ where d is the Euclidean distance between centroids and α is set to 3° and (iii) the MSE of the plane in c is lower than 0.001.

As shown in Fig. 4, the resulting mask from tile-wise Region growing will be coarse thus as a final step we refine

the tiles on the border by checking for all their pixels if the distances to the plane solution given by all points in the grown region is less than 9 times the plane MSE. Once the ground mask is obtained, we can now check which ROI features are inside it and use these for plane fitting.

D. Multi-RANSAC plane fitting

Given a set of 3D points around the target, we can estimate the ground plane parameters by fitting a plane to these. We use a RANSAC loop with MSAC criteria to reject depth outliers and also because there can be multiple planes within the point cloud. Moreover, we have experienced that the plane that maximizes the number of inliers is often a plane traversing multiple surfaces. Thus, we use a Multiple model RANSAC scheme that finds the N largest planes that have significantly different plane parameters between them, i.e. angle between normals must exceed 15° and they should not share more than 80 $\%$ of their inliers. Plane solutions that have less inliers than a fraction of the number of inliers of the largest plane solution are discarded. Finally, we select the flattest plane from this set of plane solutions based on the plane normals. Another advantage of Multi-RANSAC is that we get a depth uncertainty estimate by ray-casting the multiple plane solutions and fit a Gaussian to the depth estimates.

E. Temporal fusion

The ground mask can be under-segmented or the ground plane can be temporally covered (e.g. trees or mountains) resulting in few ROI points and in turn noisy plane estimates. Therefore, we found beneficial to use planes and depth estimates from past frames. To do so, we accumulate in a buffer the point inliers given by the plane solution – estimated in the last section. If the number of points in the buffer exceeds a certain number we remove the oldest points. Using these accumulated points we run the Multi-RANSAC plane fitting once more and store the plane solution for the next frames. This way if the ground mask fails or number of ROI is too low in the next frames we reuse this plane to raycast target depth. Additionally, since the ground mask can also be over-segmented and lead to a wrong estimated plane we gate the incoming points by checking if they are less than a given distance from the past plane solution. This distance threshold is set based on MSE of the plane that fits all accumulated points. This gating enforces temporal consistency assuming that the ground plane does not change rapidly.

VI. Experiments

We implemented two versions of TRADE. For offline evaluation, we used the full TRADE pipeline, depicted in Fig. 2, whereas for real-time applications we deployed a lighter TRADE version on a Voxl 2 board. These are described respectively in Section VI-B and VI-C.

A. Synthetic Datasets and Evaluation Methodology

There are many tracking datasets [2] but to our knowledge none includes ground-truth depth, camera poses and IMU. Therefore to evaluate object 3D localization and tracking we created 3 scenes using Unreal Engine 4, shown in Fig. 5: *Rocky desert*, *City* and *Mountain*. Each scene has one object target, which is moved along a Spline. Using the Airsim [23] plugin we set drone flights to follow the target and collected IMU at 200 Hz and 1024×1024 px images at 30 fps using a downward-looking camera with 110° of FOV. One sequence was recorded per scene, and for each of these sequences we generated 5 noisy sequences by adding Shot noise to the images and IMU noise based on the MPU9250 noise model. Camera pose was obtained by running xVIO [24] on each noisy sequence. Additionally for evaluating tracking we randomly shifted the initial tracking frame within 10 frames and we initialized the tracker with the respective ground-truth bounding box coordinates perturbed by an offset randomized within an interval of 5 pixels.

For the *Rocky desert* the drone flew horizontally 90 m at 16 m of altitude above ground, for the *City* the drone flew horizontally 20 m at 32 m of altitude above ground and for the mountain the drone flew horizontally 82 m at 30 m of altitude above ground with a 65° yaw turn after 50 m.

B. Offline Implementation Details

We used DIMP [1] as our object tracker in our full TRADE implementation. In terms of ground feature tracking, the ROI structure for detecting features was set by adding margins to the target bounding box. These margins were 2.0 times the minimum side of the bounding box for both *City* and *Mountain* and 1.0 for the *Rocky desert* where the ground is more textured. For each ROI tile, from the 3×3 grid, we set the maximum number of features per tile: 25. For temporal fusion we set the maximum number of points to 1000. For multi-RANSAC plane fitting we set the inlier threshold to 0.5 m and use a total of 4 solutions with at least half the number of inliers of the top solution. For ground plane segmentation in full TRADE, the point cloud is split into 20×20 tiles and for its region growing, we use a total of 8 seeds. Even though, the plane segmentation is not optimized, the algorithm is an adaptation of [22] which can run at 300 fps on a single CPU thread.

C. Real-time Implementation and Target Following Details

A lighter TRADE version was deployed on a Voxl 2 CPU. For this, we used STARK [25] instead of DIMP. STARK runs at about 5-10 fps on this CPU using OnnxRuntime while sharing the CPU cores with VIO [24]. Thus, the object localization module (shown in Fig. 2) was moved to a separate C++ thread to keep up with the camera frame-rate. This thread performs only ROI Feature Tracking and robust plane fitting with temporal fusion to estimate and update the ground plane model. The single-image depth and ground plane mask modules were not included in this light-version as the scenes in our real-world experiments are approximately planar.

Fig. 5: Scenes used in this work. *Left*: Rocky desert with several cars identical to the target car which moves through the rock field following a spline (shown as a white line). The drone flight is set to follow the target. *Center*: City. The drone takes off from the building's rooftop and flies along its edge to follow the white truck. *Right*: Mountain with icy lake where a tank moves along the dirt-road

As shown in Fig. 1 we used a FLIR Boson thermal camera mounted with a 45° pitch on a quadcopter. For target following, waypoints are continuously sent to the PX4 flight controller based on the 3D object filtered location. This is given by the Kalman Filter that is used for trajectory prediction (in Section IV). This results in a smooth drone path despite unstable bounding boxes. The waypoints are send to keep the same drone altitude and yaw while a horizontal offset is added to keep the object centered in the FOV.

D. Results: Depth Estimation

To evaluate depth estimation, we disabled tracking and used the ground truth bounding boxes to remove the influence of tracking errors. Each sequence was repeated 5 times due to RANSAC stochastic behaviour. Table I and II show the RMSE of depth estimation for all frames in the 5 runs. Fig. 6 shows the average depth error per frame.

First, Table I shows the results for ground depth estimation by ray-casting the ground truth bounding box center. The ground truth depth is given by the terrain surface. ROI feature depth estimation was performed using either VIO or ground truth pose to separate the effect of camera pose errors. The results are overall better on the *Rocky desert* since the flight altitude is lower and the ground is more consistently textured. Temporal fusion improves overall the results especially when combined with the ground plane mask. On the *Rocky desert*, our multi-RANSAC plane fitting is already successful in rejecting depth measurements from the rocks. Whereas on the *City* the mask is necessary to reject measurements from the building which is about 30 m tall. On the *Mountain*, combining the mask with temporal fusion also leads to more robustness, as shown in Fig. 6. Results using VIO poses are significantly worse than using the ground-truth pose, indicating that this filter-based VIO needs to be improved or fine-tuned for such high-altitude flights.

Instead of ground plane estimates, Table II shows the results of estimating the object's depth simply by sampling the Single-Image depth map after affine transformation, which we estimate as described in Section V-B with and without RANSAC. Here we use the ground-truth depth of the target to directly evaluate the accuracy. We have found the performance to be inconsistent throughout the sequence. While using RANSAC improves the RMSE on the *Rocky desert*, on the *City* it makes it worse by limiting the RANSAC samples to the building. This can be improved by using a more sophisticated sampling scheme. But overall the performance is far from the ground plane estimates.

Temporal fusion	Ground mask	Rocky desert	City	Mountain
✗	✗	0.16 (1.64)	30.4 (30.3)	3.05 (5.20)
✗	✓	0.17 (1.55)	3.74 (4.07)	1.83 (5.04)
✓	✗	**0.13** (1.39)	31.1 (31.0)	3.00 (3.83)
✓	✓	**0.13 (1.26)**	**1.46 (3.5)**	**0.55 (2.98)**

TABLE I: RMSE of ground depth estimates [m] using either ground-truth camera pose or VIO [24] in brackets. Ground segmentation and temporal fusion were switched on-off.

RANSAC	Rocky desert	City	Mountain
✗	4.69	14.77	4.54
✓	1.66	26.72	4.60

TABLE II: RMSE of depth estimates [m] using Single Image depth. Ground-truth camera pose was used for feature depth estimation.

E. Results: Object Tracking

Object tracking performance is evaluated between TRADE and DIMP on 10 Monte Carlo samples per sequence. Results are shown in Fig. 7 as the Intersection-over-Union (IoU) between the ground-truth and estimated bounding boxes. On the *Rocky desert*, DIMP fails all 10 runs due to the *target switching* shown in Fig. 1. There's also partial occlusion, which causes the DCF to loose temporally tracking. This appears as the lowest average IoU for TRADE in Fig. 7.

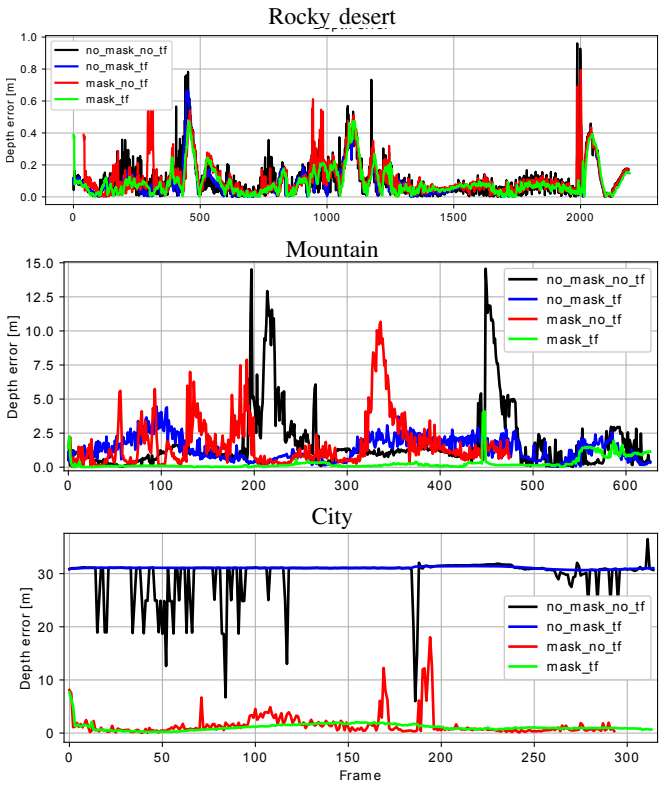

Fig. 6: Depth error per frame averaged over 5 runs, corresponding to Table I. *no_mask_no_tf* stands for disabled ground mask and temporal fusion whereas *mask_tf* stands for full TRADE.

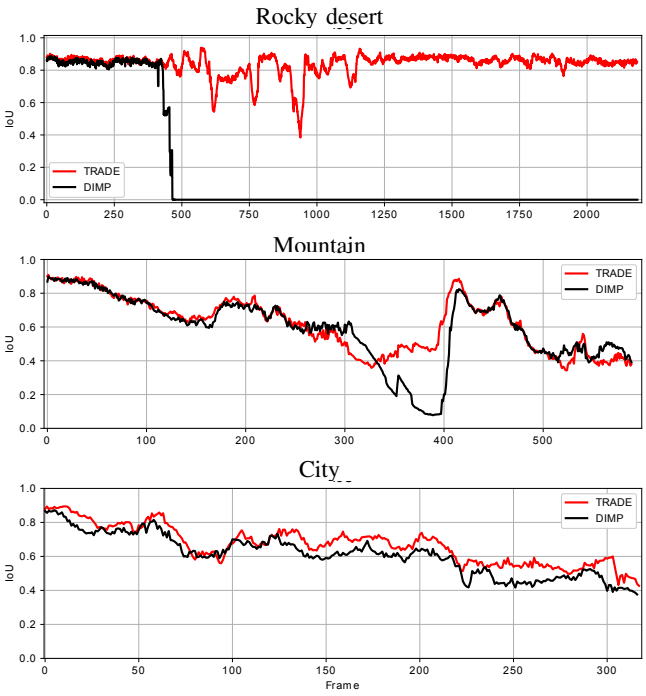

Fig. 7: Tracking accuracy as bounding box IoU per frame averaged over 10 runs. Camera pose was estimated using VIO [24]

On the *Mountain*, DIMP also suffers from *target switching*, shown in Fig. 8. whereas TRADE is constrained by the trajectory seen as magenta on Fig. 8. On the *City*, the differences are not that significant. For more details check our supplementary video.

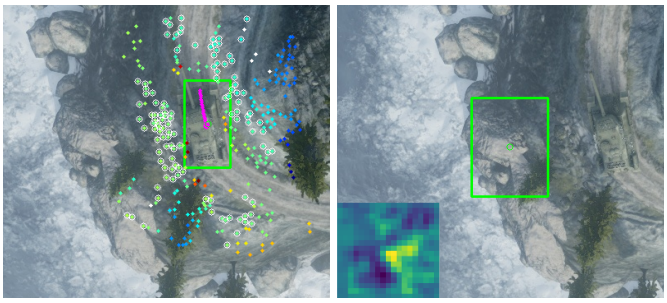

Fig. 8: Examples of challenging tracking. *Left:* TRADE using predicted trajectory and ground feature depth estimates (color-coded based on depth). Features with a white circle are the plane fitting inliers. *Right:* On the same frame DIMP [1] performs *target switching* due to multi-peak tracker's response shown at the corner.

VII. Conclusion and Limitations

The combination of all 3d localization components in TRADE shows improved robustness to complex terrain. We also demonstrate a practical approach to improve tracking robustness by incorporating the trajectory predictions. However a few concerns remain. The tracking performance is sensitive to the fixed process and measurement noise of the Kalman filter. A more adaptive parameterization as in [10] is desirable. Using a dedicated single-image depth estimation process introduces a significant computational cost. This can be alleviated by using only when necessary, e.g., during initialization. We demonstrated a simple but effective target-following operation using TRADE onboard. This can be further improved by taking into account motion planning, control and the camera pitch and FOV.

VIII. Acknowledgments

The research was funded by the Combat Capabilities Development Command Soldier Center and Army Research Laboratory. The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004). ©2022 California Institute of Technology.

References

[1] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *ICCV*, pp. 6182–6191, 2019.

[2] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, J.-K. Kämäräinen, M. Danelljan, L. Č. Zajc, A. Lukežič, O. Drbohlav, *et al.*, "The eighth visual object tracking vot2020 challenge results," in *ECCV*, pp. 547–601, Springer, 2020.

[3] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "Know your surroundings: Exploiting scene information for object tracking," in *ECCV*, pp. 205–221, Springer, 2020.

[4] C. Mayer, M. Danelljan, D. P. Paudel, and L. Van Gool, "Learning target candidate association to keep track of what not to track," in *ICCV*, pp. 13444–13454, 2021.

[5] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *International Conference on Image Processing (ICIP)*, pp. 3645–3649, IEEE, 2017.

[6] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *ICCV*, pp. 941–951, 2019.

[7] P. Zhang, J. Zhao, C. Bo, D. Wang, H. Lu, and X. Yang, "Jointly modeling motion and appearance cues for robust rgb-t tracking," *IEEE Transactions on Image Processing*, vol. 30, pp. 3335–3347, 2021.

[8] M. Henein, J. Zhang, R. Mahony, and V. Ila, "Dynamic slam: the need for speed," in *ICRA*, pp. 2123–2129, IEEE, 2020.

[9] J. Amirian, J.-B. Hayet, and J. Pettré, "Social ways: Learning multi-modal distributions of pedestrian trajectories with gans," in *CVPR Workshops*, 2019.

[10] H. Coskun, F. Achilles, R. DiPietro, N. Navab, and F. Tombari, "Long short-term memory kalman filters: Recurrent neural estimators for pose regularization," in *ICCV*, pp. 5524–5532, 2017.

[11] X. Zhao, F. Pu, Z. Wang, H. Chen, and Z. Xu, "Detection, tracking, and geolocation of moving vehicle from uav using monocular camera," *IEEE Access*, vol. 7, pp. 101160–101170, 2019.

[12] X. Wang, J. Liu, and Q. Zhou, "Real-time multi-target localization from unmanned aerial vehicles," *Sensors*, vol. 17, no. 1, p. 33, 2017.

[13] V. Dobrokhodov, I. Kaminer, K. Jones, and R. Ghabcheloo, "Vision-based tracking and motion estimation for moving targets using small uavs," in *American Control Conference*, 2006.

[14] D. B. Barber, J. D. Redding, T. W. McLain, R. W. Beard, and C. N. Taylor, "Vision-based target geo-location using a fixed-wing miniature air vehicle," *Journal of Intelligent and Robotic Systems*, vol. 47, no. 4, pp. 361–382, 2006.

[15] H. Zhang, G. Wang, Z. Lei, and J.-N. Hwang, "Eye in the sky: Drone-based object tracking and 3d localization," in *ACM International Conference on Multimedia*, pp. 899–907, 2019.

[16] C. Yuan and G. Medioni, "3d reconstruction of background and objects moving on ground plane viewed from a moving camera," in *CVPR*, vol. 2, pp. 2261–2268, IEEE, 2006.

[17] T. Liu and Y. Liu, "Moving camera-based object tracking using adaptive ground plane estimation and constrained multiple kernels," *Journal of Advanced Transportation*, vol. 2021, 2021.

[18] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3d bounding box estimation using deep learning and geometry," in *CVPR*, pp. 7074–7082, 2017.

[19] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *Transactions on PAMI*, 2020.

[20] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *ICCV*, pp. 3828–3838, 2019.

[21] I.-S. Kweon, M. Hebert, E. Krotkov, and T. Kanade, "Terrain mapping for a roving planetary explorer," in *ICRA*, 1989.

[22] P. F. Proença and Y. Gao, "Fast cylinder and plane extraction from depth cameras for visual odometry," in *IROS*, 2018.

[23] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and service robotics*, pp. 621–635, Springer, 2018.

[24] J. Delaune, D. S. Bayard, and R. Brockers, "Range-visual-inertial odometry: Scale observability without excitation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2421–2428, 2021.

[25] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10448–10457, 2021.