Integration of 3D Knowledge for On-Board UAV Visual Tracking

Stéphane Vujasinović*, Stefan Becker, Timo Breuer, Norbert Scherer-Negenborn, and Michael Arens
Fraunhofer Center for Machine Learning and Fraunhofer IOSB, Ettlingen, Germany

1 Abstract

Visual tracking from an unmanned aerial vehicle (UAV) poses challenges such as occlusions or background clutter. In order to achieve more robust on-board UAV visual tracking, a pipeline combining information extracted from a visual tracker and a sparse 3D reconstruction of the static environment is introduced. The 3D reconstruction is based on an image-based structure-from-motion (SfM) component and thus allows to utilize a state estimator in a pseudo-3D space. Thereby improved handling of occlusion situations and background clutter is realized. Evaluation is done on prototypical image sequences captured from a UAV with low-altitude oblique views. The experimental results demonstrate the benefit of the proposed approach compared to only relying on visual cues or using a state estimation in the image space.

2 Introduction

UAV-based vision systems have drawn increasing attention in recent years with many applications, such as video surveillance, aerial photography, and visual object tracking. The latter application is the task of estimating the location of an object in an image sequence. When an additional state estimator is utilized, the tracking pipeline is referred to as detection-by-tracking, and without as tracking-by-detection [1].

Many current state-of-the-art visual tracker follow the tracking-by-detection paradigm. The estimated location of the target is solely based on the maximum-likelihood estimate, inferred from comparing an appearance model of the target with a small search region. For close-range tracking scenarios, adding a state

*email: stephane.vujasinovic@iosb.fraunhofer.de
estimator often leads to an under-performance, due to the complex problem of filter tuning for these image sequences [2]. However, despite the general progress of visual trackers, these approaches are usually not optimal or suffer from particular limitations when being applied on-board of a UAV. The main challenges for UAV-based visual trackers arise due to specific views of captured scenes. For instance, in image sequences captured from a low-altitude oblique view, tracked objects often get occluded by structures from the environment such as trees or buildings. Further, compared to a close-range frontal perspective, the size of the tracked objects or rather the object to image size ratio is relatively small, as illustrated in figure 2.

To better deal with the challenges of low-altitude UAV views, we propose a visual tracking pipeline that combines a visual tracker with a sparse reconstruction of the static environment and a state estimator. Thanks to the 3D reconstruction, the pixel location estimated by the visual tracker can be associ-
ated with a 3D location. Thus, the proposed approach enables us to employ a recursive Bayesian filter in the reconstructed 3D environment shifting toward a detection-by-tracking pipeline.

Utilizing the 3D environment offers several benefits. Firstly, a more faithful transition model of the object dynamics can be incorporated in the state estimator and the tuning of the filter is also simplified. Secondly, a false association between the target and a distractor can be prevented with a form of gating. Thirdly, occlusion occurrences can be adequately predicted. The proposed approach can be decomposed into three main components: A visual tracker for modeling the appearance and locating the target, a SfM component for generating a 3D sparse reconstruction, and lastly, a state estimator working in the 3D reconstruction. All these components can be exchanged with alternative solutions for further improvements.

In this paper, we consider the state-of-the-art visual trackers ATOM [5] and DiMP [6] for appearance modeling and generating the observations for the state estimator. The SfM-based 3D scene reconstruction is performed by COLMAP [7,8,9], and the state estimator is a particle filter. For evaluation, the publicly available dataset AU-AIR [4] is employed. The dataset is carefully further edited to best reflect prototypical occlusion situations from a low-altitude oblique UAV perspective.

The paper is structured as follows. In section 3 an overview of related work on visual tracking and visual tracking benchmarks is provided. The complete system and the individual components are explained in section 4. In section 5 we give a detailed description of the edited dataset used, analyze quantitative and qualitative results, and present the benefits of the design choices made. Finally, a conclusion is given in section 6.

3 Related Work

In recent years, great progress regarding visual tracking has been made, thanks to the abundant benchmarks available [2,3,10,11,12,13,14,15,16,17,18]. Most of them are designed towards evaluating trackers on eye-level perspectives, resulting in state-of-the-art visual trackers following the tracking-by-detection paradigm. Currently, three tracker designs prevail on those benchmarks. Firstly, the Discriminative Correlation Filter (DCF) approach [19,20,21,22,23,24,25]. Secondly, the Siamese based approach [26,27,28,29,30,31]. And lastly, a recent alternative to both styles are trackers inspired by correlation filter but employ instead a small convolutional neural network learning the appearance model of the target [5,6]. In all 3 design choices, the main difference lies in how they learn an appearance model of the target. The latter style is explored in this paper.

However, as above-mentioned, these trackers are tailored to scenarios presenting eye-level perspectives, but on-board UAV perspectives present particular challenges, consisting mainly of occlusions, clutter situations, and tracking small objects. Since UAVs equipped with cameras have expanded in usage conjointly
with the number of applications that they provide, several benchmarks have been introduced \cite{32, 33, 34, 35}, ranging from low to high altitudes, and propose either an oblique viewpoint from the UAV or a top-down viewpoint. Most participating trackers in the latter benchmarks use state-of-the-art trackers presented in the former benchmarks, that have been adapted to some extent for the UAV perspectives. Nevertheless, none of the adapted trackers participating in the UAV benchmarks attempt to utilize 3D information. The main reasons are that such information is not provided and an image-based reconstruction would prove to be difficult for most sequences. In contrast to the popular UAV benchmarks for tracking, the AU-AIR dataset \cite{4} introduced recently, is oriented toward object detection from a UAV viewpoint. It offers sequences that capture typical traffic on a roundabout, as would a surveillance drone for traffic monitoring. These image sequences are ideal, up to a certain degree for reconstructing the 3D environment of the scene.

For applications such as autonomous driving, it is common to follow the detection-by-tracking paradigm and to estimate the state of a tracked object in a 3D system of reference. Typically the 2D image is mapped into a 3D system of reference using stereo-cameras or visual odometry. An example of an application is pretended in \cite{36}, where the authors propose to use object detection, stereo, visual odometry, and optical scene flow and a Kalman filter to enhance tracking performance on the KITTI benchmark \cite{37, 38}. A follow up to this study is \cite{39}, which reconstructs the static scene and the object target in 3D, allowing to shift from tracking in 2D space toward a tracking in 3D space. In addition, the reconstructed target is associated with a velocity, inferred from the optical flow of the target, which is latter associated with tracklets, enabling the authors to overcome occlusion situations and missing detections. Independent of the way the mapping to reference is realized, the overall tracking performance is enhanced by estimating the state of the target in a 3D system of reference, rather than in the image space. We believe, that tracking applications such as visual tracking from a UAV, could also benefit from a shift towards the detection-by-tracking paradigm by incorporating 3D information.

In this paper, we apply a model-free single object visual tracker, which means that the tracker can only track a single object and starts with a blank appearance model, unlike a detection pipeline. Regardless of the method used for training the tracker (i.e. offline, online), an appearance model of the target is used to locate the target in the image space. Important for enabling the projection of the 2D coordinates in a 3D space, is a working mapping reference system. Here an SfM component is used to compensate the ego-motion of the UAV and to reconstruct the static environment. Since we do not reconstruct dynamics objects in the 3D referential, we simulate the target with a particle filter.
4 Visual Tracking Pipeline

The proposed visual tracking pipeline consists of three main components. In the following section, each component is described in detail.

![Visual Tracking Pipeline Diagram]

Figure 3: The architecture of the proposed visual tracking pipeline. The visual tracker outputs a similarity score map based on the search area, an initial position for the target and the bounding box. The similarity score map and pseudo-depth map of the image are processed by the 3D Context component, inferring a new position for the target in the image.

4.1 Appearance Modeling

In this work we rely on two state-of-the-art visual trackers, ATOM [5] and DiMP [6]. The chosen trackers achieve top ranks against numerous participants on general visual tracking benchmarks [2, 3, 11, 12, 15, 17, 32].

ATOM [5] is made out of three components: Firstly, a feature extractor which is a pre-trained neural network e.g. ResNet18, ResNet50 [40] that extracts salient features. Secondly, a classification component composed of a two-layer convolutional neural network, trained during the tracking process to learn the appearance model of the target, enabling it to propose an initial estimation of the location and bounding box of the target in the image. Thirdly, the target estimation component trained offline, based on the IoU-Net [41], designed to refine the initially proposed bounding box and position.

DiMP [6] is a successor of ATOM and builds on the same components. Essentially, the main difference lies in the extension of the classification component. By employing a different strategy for the initialization of the weights for the classification network, more efficient weights are provided for the start off. Moreover, the online learning of the appearance model is refined to allow faster and more stable convergence.
During the tracking cycle, salient features from the search area are extracted through a feature extractor. Based on the extracted features, a similarity score map is inferred, used in estimating an initial bounding box and position for the target. Afterward, a more accurate position and refined bounding box are estimated through the bounding box regressor.

A block diagram outlining the main components of a standard pipeline for single object visual tracking is presented in figure 4. During an incoming frame, a search area is firstly created to delimit the possible positions of the target in the frame. The positioning and the size of the search area depend on the previous estimated position and size of the target. The backbone neural network i.e. ResNet-18 or ResNet-50 extract salient features from the search area. A similarity score map, reflecting how good the features match with the learned appearance model, is calculated based on the features extracted from the search area and the appearance model of the target. The highest score in the similarity score map is designated as the position of the target, and the target estimation module i.e. IoU-Net is used to identify the best fitting bounding box and to refine the estimated position.

During tracking the appearance model is updated based on past extracted features. Updates on the appearance model occur every 10 newly valid added frames on ATOM and every 15 frames on DiMP. A search area is considered to be valid in case the tracker has no difficulties in identifying a single target in conjunction with a high enough similarity score. The tracker deals with distractors by recognizing more than one strong similarity peak, switching the training strategy, where the tracker immediately updates the appearance model with a higher learning rate.

### 4.2 Particle filter

For estimating the state of the target in the reference system, a particle filter is used. The particle filter estimates the posterior state density of the target \( p(x_t) \) based on, a transition model \( f \), the current observation \( z_t \), and the prior state distribution \( p(x_{t-1}) \). The state describes the position and velocity of the target for each time step \( t \) in the studied space. In order to estimate the posterior state probability density function \( p(x_t) \), the particle filter uses particles, where each
particle denotes a hypothesis on the state. Particles from the prior probability density function \( p(x_{t-1}) \) are propagated through a transition model e.g. constant velocity model. In a second step, weights \( w_i \) at time step \( t \) are assigned to \( N \) particles, \( i \in [1, ..., N] \), mirroring how strong particles match with the current observation \( z_t \).

Let \( p(x_t|z_{1:t}) \) represent the probability distribution of the posterior state given the current observation \( z_t \) in equation 1. Each particle has a corresponding weight \( w_i \) at time \( t \), for \( i \in [1, ..., N] \) with \( N \) being the total number of particles. \( x_i^t \) is the hypothesis on the state of the \( i \)-th particle at time \( t \), and \( x_t \) the state estimation at time \( t \). \( \delta \) is the Dirac delta function. The weights are normalized such that \( \sum_i^N w_i = 1 \). Particles that better match the current observation \( z_t \) are weighted higher than particles that are dissimilar to it. In our case, the similarity score map of the visual tracker is used to weight the particles.

\[
p(x_t|z_{1:t}) \approx \sum_{i}^{N} w_i \delta(x_t - x_i)
\]

Following the weighting of the particles, a resampling step is performed whenever \( N_{eff} \) presented in equation 2 is below a certain threshold. The resampling allows the particle filter, to filter out unnecessary particles that have low weights, and create new particles based on the stronger weighted particles, allowing a refined approximation of \( p(x_t|z_{1:t}) \).

\[
N_{eff} = \frac{1}{\sum_i^N (w_i)^2}
\]

4.3 Sparse Reconstruction

In order to map the 2D image space to a 3D reference system, a structure-from-motion component (SfM) is used. In this paper, we rely on COLMAP [7][8][9], which generates a sparse reconstruction of the static environment using an image-based SM reconstruction. Figure 5a displays a point cloud constructed by COLMAP on one of the image sequences used in this paper. The static reconstruction of the scene and the positions of the UAV (i.e. the camera) are calculated beforehand being used by our tracker.

As shown in figure 5a, the initial reconstruction contains outliers and noise near the reference frame of the static reconstruction. The noise near the reference frame is due to elements badly triangulated in space, thus estimated close to the UAV (i.e. the camera). The outliers are points which have been corrected placed in the reconstruction but do not form a large enough group to represent a structure reliably. To reduce the number of misleading points in the 3D reconstruction, we filter out the points that are close to the reference frame and the outliers, as presented in the figures 5b and 5c.
Figure 5: (a) Left reconstruction is a rough point cloud structure given by COLMAP. (b) In the middle reconstruction, the noise near the frame of reference is highlighted in red. (c) Right reconstruction has the noise near the frame of reference discarded and highlighted in red are identified outliers.

Thereby a filtered version of the point cloud is obtained as presented in figure 6 for the two sequences used in this paper. In addition, a plane \((O_g, \vec{x}_g, \vec{y}_g)\), representing the ground in the 3D reconstruction is fitted in each point cloud. Figure 7 shows a filtered reconstruction with a fitted ground plane estimation \((O_g, \vec{x}_g, \vec{y}_g)\) in blue.

Figure 6: Filtered point cloud for both sequences composing the dataset. (a) is a reconstruction of sequence 0 and (b) of sequence 1.

Figure 7: A filtered sparse reconstruction with the ground plan estimation in blue.

Based on the sparse reconstruction, we build pseudo-depth maps, which encode for each pixel the distance between the current position of the UAV and the visible points of the sparse reconstruction for the corresponding viewpoint. Figure 8 displays a pseudo-depth maps on the right of the left image. For every frame in both image sequences, a corresponding pseudo-depth map is
constructed, thus achieving the desired mapping from the image space to a 3D space.

Figure 8: The right image, is the pseudo-depth map reconstructed using the point cloud of the observed scene and the viewpoint of the frame, shown on the left side.

4.4 Tracking Cycle and Occlusion Handling

During initialization, the visual tracker learns an appearance model of the target, and at the same time the position of the target in the image is projected onto the estimated ground plane of the 3D reconstruction. The projected position expressed in 3D coordinates is considered the initial position for state $x_{-1}$, but no velocity is assigned to this state at the time. In the following frame, the visual tracker provides the position of the target on the current frame. In addition, particles are generated uniformly over the projected surface of the current search area on the ground plane estimation of the 3D reconstruction. The particles are weighted accordingly to the similarity score map of the visual tracker, and an estimation of the position of the target in 3D is achieved. Moreover, the current 3D position along with the previous 3D position are used to determine an initial velocity in the 3D space for the current state. Thus, an initial state $x_0$ for the particle filter is defined, with a constant velocity and particles scattered around the target position on the ground estimation. In the following frame, the particle filter can be used to predict the position of the target in the 3D reconstruction. During prediction, a constant velocity model is adopted, and a Gaussian noise term is added to diversify the particle position and velocity hypothesis. The particles are constrained to move only on the ground estimation of the 3D reconstruction.

After initialization, the tracker enters an online tracking phase. Figure 3, displayed at the beginning of this section presents the essential architecture of the framework. On an incoming frame, the visual tracker defines a search area and produces a similarity score map along with estimating an initial position and bounding box of the target in the image space. The similarity score map and the pseudo-depth map of the current frame are transmitted to the 3D Context component. The 3D Context component is used to estimate the position of the target in the 3D reconstruction, to distinguish the target from distractors, and to recognize occlusions. The new 3D position of the target $(x_w, y_w, z_w)$,
expressed in the pseudo-3D world reference system is projected back onto the image space, corresponding to the new estimated position \((x_i^n, y_i^n)\) of the target in the image.

Figure 9: Architecture of the Context 3D component. The prediction of the particle filter and the pseudo-depth map are used to identify particles that are out of the ground plane estimation. Particles are also clustered depending on their position in the image to identify potential groups such as the target and distractors. In case the target is considered occluded, then the 3D coordinates of the prediction \((x_p^w, y_p^w, z_p^w)\) are considered and re-projected in the image as \((x_i^n, y_i^n)\). If the target is not identified as occluded, then an update on the particles and a resampling is performed before clustering the particles. Following the determination of the cluster representing the target, the 3D coordinates of the cluster attached to the target \((x_u^u, y_u^u, z_u^u)\) are re-projected in the image as \((x_i^n, y_i^n)\).

Figure 9 displays the different building blocks of the 3D Context component. It contains the particle filter, which estimates the target state in the sparse reconstruction i.e. 3D position and 3D velocity. As stated above, during prediction, a constant velocity model is used, and to model the uncertainty in position and velocity a Gaussian noise term is added to the velocity during the prediction stage of the particle filter. The additive noise is important as it allows us to explore a unique hypothesis for each particle, by dispersing them around the potential position of the target and allows multiple assumptions regarding the velocity of the target. The predicted particles are subsequently used by the occlusion identifier to determine if the target is hidden by a structure in the sparse reconstruction. Although the object is visible, a small percentage of particles are uniformly re-distributed across the projected search area in the sparse reconstruction. This re-distribute of particles over the search area is needed to maintain a multimodal distribution. If no re-distribution is performed, the particles would form a clump, which would imply that only a small portion of the similarity score map is used to weight the particles. Regarding the occlusion identifier, the input elements are the predicted particles in 3D and the pseudo-depth map corresponding to the current image. By comparing the
depth value of the predicted particles in the UAV (i.e. camera) reference system and the depth value in the pseudo-depth map, we can identify particles that are hidden by elements present in the 3D reconstruction. If the number of particles identified as occluded exceeds a threshold i.e. 50%, the target is considered hidden. Similarly, the target is automatically considered occluded if the resulting similarity score map of the appearance model is flat and widely spread over the search area.

The weighting of the particles is based on projecting their 3D positions onto the similarity score map of the appearance model. Thereby, particles that are located near a high peak in the similarity score map are weighted higher. This is followed by a resampling step of the particles through stratified resampling. The particle filter allows multimodal tracking, thus avoids a maximum likelihood decision, where only the highest score value in the search area is considered. This prevents quick jumps to distractors as presented in figure 11.

We cluster particles in the image space, but compare the positions of the clusters \((x^c_w, y^c_w, z^c_w)\) with the predicted position of the target \((x^p_w, y^p_w, z^p_w)\) in the 3D reconstruction, to identify the cluster describing the target. The cluster closest to the predicted position is considered to be the cluster corresponding to the target, thus the position of the target is \((x^u_w, y^u_w, z^u_w)\) in the 3D system of reference. The coordinates in 3D are then projected in the image space, giving us the image coordinates \((x^n_i, y^n_i)\), which are afterwards used to update the appearance model of the visual tracker. This allows us to avoid adding misleading training samples e.g. when the tracker mistakes the target with a distractor, during the update of the appearance model.

Figure 10: The small green box represents the belief of the visual tracker and the red box the belief of the visual tracker with the particle filter. In this scenario, the similarity score map has three peaks. Due to the maximum likelihood approach of the visual tracker, it mistakes a distractor with the target. Whereas the visual tracker with particle filter manages to stay on the target, even though the highest similarity score is attached to a distractor.
5 Results

5.1 Dataset

We use the AU-AIR dataset for evaluation because of the real-life scenario it provides for traffic monitoring. The dataset reflects prototypical outdoor situations captured from a UAV. A total of 8 individual image sequences make up the dataset, decomposed in 32,823 frames, taken from a low flight altitude ranging from 10 to 30 meters, and are taken under different camera angles ranging from 45 to 90 degrees. For each frame, the dataset provides recording time stamps, GPS coordinates, altitude, inertial measurement unit (IMU) data, and the velocity of the UAV. A criterion in favor of AU-AIR is the low range altitude flights with an oblique point of view towards the observed scene it provides. Thus, objects have a higher chance to be occluded from this perspective rather than from a top-down perspective.

![Sequence 0](image1.png)
(a) Sequence 0.

![Sequence 1](image2.png)
(b) Sequence 1.

Figure 11: Examples of images from the two sequences extracted from the AU-AIR dataset.

A focus for this study was to design a module for trackers that would enable the tracker to overcome occlusion situations using a 3D reconstruction of the static scene. Thus, the dataset includes the desired challenges of occlusions, which are required to analyze the benefit of incorporating 3D scene knowledge. For this purpose, two video sequences have been distilled from AU-AIR, composed of two sequences. The carefully processed dataset is referred to as AU-AIR-Track in the following. AU-AIR-Track is composed of two sequences, named sequence 0 and sequence 1 presenting only oblique viewpoints from the UAV and have enough movement to enable a 3D reconstruction of the scene using SfM. The former sequence is made out of 887 frames, corresponding to approximately 3 minutes, and includes 62 annotated objects for visual tracking. The latter sequence is made out of 512 frames, which is equivalent to 1 minute and a half and contains 26 annotated objects. Figure 11 displays a preview on example images from both sequences. As a result, the main challenges for this new dataset are the constant camera motion, the low image resolution, the
presence of distractors, and most importantly many occlusion situations.

Figures 12a and 12b show the distribution of ground truth bounding box locations for both sequences. The value of each pixel denotes the probability of a bounding box to cover that pixel over an entire sequence. It can be seen that most objects follow the underlying scene structure, i.e the road. It should be noted that no semantic information of the scene is used during the tracking process.

![Figure 12: Distribution of ground-truth bounding boxes.](image)

From the 63 possible targets present in sequence 0, 45 targets undergo an occlusion, and 24 out of the 26 targets in sequence 1. Figure 13 shows the distribution of ratio between the duration where the target is hidden in respect to the total duration where the target is in the scene.

![Figure 13: Distribution of occlusion ratios for sequence 0 (left) and sequence 1 (right) of AU-AIR-Track.](image)

Since AU-AIR annotations are designed for object detection, we adapted them for visual tracking and added occlusion annotations. To this end, identi-
fication numbers have been assigned to the objects. In the refined annotations\textsuperscript{1}, only moving objects are annotated.

### 5.2 Evaluation Metrics

Similar to long-term tracking, the tracked object can disappear and reappear. Thus, no manually re-initialization is done when the tracker loses the target. To measure the performance, we utilize commonly used long-term metrics, tracking precision $Pr$, tracking recall $Re$ and tracking F1-score, introduced in \cite{13} and used in VOT \cite{11, 2}. Accuracy and Robustness are not used in this evaluation, since these metrics are better suited for short term tracking with short occlusion periods.

Let $G_t$ be the ground truth target bounding box, and $A_t$ the bounding box estimation given by the tracker at frame $t$. $\theta_t$ is the prediction certainty or in our case the maximal score given by the tracker regarding its confidence on the presence of the target in the current frame $t$ and $\tau_\theta$ is a threshold applied to $\theta_t$. If the target is absent or partially or fully occluded, $G_t = 0$, and similarly, if the trackers predicts a target with a confidence $\theta_t$, that is below $\tau_\theta$, $A_t = 0$. Furthermore, $N_g$ is the number of frames where $G_t \neq 0$ and $N_p$ the number of frames where $A_t \neq 0$. Lastly, $\Omega(A_t(\theta_t), G_t)$ is the IoU between $G_t$ and $A_t$.

$$
Pr(\tau_\theta) = \frac{1}{N_p} \sum_{t \in \{t: A_t(\theta_t) \neq 0\}} \Omega(A_t(\theta_t), G_t), \hspace{1cm} (3)
$$

$$
Re(\tau_\theta) = \frac{1}{N_g} \sum_{t \in \{t: G_t \neq 0\}} \Omega(A_t(\theta_t), G_t), \hspace{1cm} (4)
$$

$$
F(\tau_\theta) = \frac{2Pr(\tau_\theta)Re(\tau_\theta)}{Pr(\tau_\theta) + Re(\tau_\theta)} \hspace{1cm} (5)
$$

The combination between tracking precision $Pr(\tau_\theta)$ and tracking recall $Re(\tau_\theta)$ into a single score is defined as the tracking F1-score \cite{13}. Similarly to the long term challenges of \cite{11, 2} the final tracking F1-score is used to rank the different tracker versions. The evaluation protocol is as follows: the trackers are evaluated on all targets present in the AU-AIR-Track. The annotated first frame of the target is used to initialize the tracker. From there the tracker outputs a prediction bounding box for every subsequent frame where the target is annotated even during occlusions, and no reset is allowed. Tracking precision, tracking recall and tracking F1-score are computed accordingly to equations 3, 4 and 5. To avoid statistical error we run an evaluation of each tracker five times on both sequences of the AU-AIR-Track. For each evaluation and for each target present in a sequence, we take the maximum tracking F1-score regardless of $\tau_\theta$ for that target. We average together the extracted tracking F1-scores maximums of each target, belonging to the same sequence and evaluation, giving

\textsuperscript{1}Upon request, visual tracking annotations can be provided with the pseudo-depth maps and point clouds of the reconstructions.
Table 1 summarizes the final tracking F1-scores for ATOM and DiMP for the original, 2D, and 3D versions. The blue color indicates the best F1-score for each version of ATOM and DiMP on sequence 0 and 1 of AU-AIR-Track. Based on these results, the 3D version displays better performance in comparison to the original and 2D versions. In figure [14] the score for individual tracked object are visualized. The 3D version with ATOM, performances approximately better on 55% targets in both sequences, and the 3D DiMP version on 45% targets.

| Version     | Tracker | Sequence 0 F1-score | std | Sequence 1 F1-score | std |
|-------------|---------|---------------------|-----|---------------------|-----|
| Original    | ATOM    | 0.621               | 0.221 | 0.584               | 0.226 |
|             | DiMP    | 0.593               | 0.247 | 0.579               | 0.249 |
| 2D Version  | ATOM    | 0.642               | 0.222 | 0.608               | 0.243 |
|             | DiMP    | 0.665               | 0.199 | 0.617               | 0.235 |
| 3D Version  | ATOM    | 0.678               | 0.194 | 0.709               | 0.169 |
|             | DiMP    | 0.722               | 0.147 | 0.711               | 0.141 |

Table 1: Averaged F1-scores from the different ATOM and DiMP versions on sequence 0 and sequence 1 of the AU-AIR-Track dataset. Results highlighted in blue correspond to the highest averaged F1-score observed for each sequence.

Based on table 1, we observe that the original version of ATOM and DiMP are less suited for the type of scenarios present in the dataset. The original trackers attain the lowest scores and the least stable results. As stated before, the original versions of the visual trackers are designed for short-term tracking from an eye-level perspective by relying on visual cues, and expecting objects with a high object-to-image ratio. Whereas, from a drone perspective most of the time the objects tracked are relatively small and have a low object-to-image ratio. Moreover, no specific module is integrated for recognizing or handling occlusions. Another limiting factor to the performance of the original trackers are compression artefacts due to the low resolution of the object area, which makes it difficult for the tracker to learn a unique appearance model.

For the 2D versions, recognition of occlusion is only possible through visual cues. This can be done by setting a minimum required similarity score as a threshold. When an occlusion occurs, the position of the target cannot be inferred through visual cues, thus the predicted position of the target given by the particle filter is used. The prediction of the particle filter is used until a high enough response in the similarity score map beyond a threshold is detected. This solution is limited since the tracker could misinterpret a fast appearance change with an occlusion. Another limitation is that only occlusions without ego-motion can be handled. Nevertheless, there is an increase in the tracking F1-score.
compared to the original version. This gain in performance is essentially due to better recognizing and handling distractors achieved through the multi-modal representation offered by the particle filter. By using this property, groups of particles can be formed at different locations and we can cluster them, allowing the tracker to distinguish the target from the distractors. Whereas, the original versions utilize a maximum likelihood approach, thus expecting the target to be located where the highest response in the score map is estimated.

Regarding the 3D versions, they achieve the best results on the AU-AIR-Track dataset as well as the best stability. The 3D versions use the sparse reconstruction of the static environment, which allows them to identify occlusions not only based on visual cues but also through depth information. Moreover, the transition model used for the particle filter in the 3D environment is more reliable to a real-world motion model, making the prediction much more stable during ego-motion. Besides, physics-based motion models in the image space, are in a way abused as general-purpose models and it is hard to adequately set the noise level parameters. However, with the assistance of the 3D reconstruction, a stable tracking is allowed regardless of camera movements, because the position is expressed in 3D coordinates in a 3D reference system rather than on pixel coordinates. As in the 2D versions, the particle filter allows a multi-modal representation, which enhances the ability of the trackers to distinguish the target from distractors.
5.4 Qualitative Analysis

In this section, we discuss selected qualitative examples in order to verify the overall viability of the different versions. To illustrate the results, an in-depth look is provided based on the DiMP versions. Figure 15 shows a sequence where the target is lost by the visual tracker in contrast to the 2D and 3D versions. The original approach uses a maximum likelihood for estimating the location of the target in the search region, causing the tracker to only consider the highest score. However, the distractor present in the search area, appears to be visually more similar to the appearance model learned by the tracker than the actual target. Hence, the tracker mistakes the target with the distractor. In contrast, the 2 other versions, using a particle filter enable the tracker to better handle the presence of distractors as illustrated in figure 10 of section 4.4.

Figure 16 displays three consecutive frames, where the 2D version loses the target due to sudden camera rotation. Only the 3D version can stay on the target thanks to the stability provided by using a 3D space to express the coordinates of the target. This sudden rotation is also responsible for the original trackers and their 2D versions to achieve a low F1-score in sequence 1, presented in figure 14 on targets 2 and 3.

Figure 15 presents an occlusion scenario where the target is hidden by a tree. Both 2D version and 3D version manage to stay on the target until reappearance. Whereas, in figure 17b only the 3D version succeeds in retrieving the target after the occlusion. The difference in results between both scenarios is due to camera motion. In the former scenario, camera motion is quite minimal, in contrast with the latter scenario where camera motion is larger. Concerning figure 17c, only the 3D version recognizes the occlusion due to the depth information provided.

Figure 16: The 3D version allows to stabilize the tracking during camera motion.
through the 3D reconstruction.

Despite the 3D versions being able to achieve remarkable results, there are some failure cases where the occlusion is not handled correctly. An example is shown in figure 18a, where the tracker identifies occlusion but recognizes at the same time a strong similarity score. In this case, the distractor is considered as the actual target not being hidden anymore, thus leading to a jump on the distractor, and discarding the previous prediction. However, this situation is not related to the usage of 3D information, but can be prevented by elaborating a different strategy for recognizing reappearances after occlusion was identified.

Another point limiting the performance of the tracker is the tuning of the particle filter as illustrated in figure 18b, where the target slows down at the intersection for a long period. Nevertheless, particles keep updating their estimated velocity to be adequate with the observation. In consequence, a velocity near zero is estimated, and during the acceleration phase of the target, the particle filter cannot match with the actual speed instantly, due to the constant velocity model used. Hence, when the car is hidden, the particle filter predicts positions belonging to a target moving slower than in reality.

![Figure 17: Different occlusion scenarios. Only the 3D version manages to overcome the occlusion in these scenarios.](image)

(a) In this scenario, the 2D and 3D DiMP versions are able to handle the target undergoing occlusion.

(b) Because of the 3D version being stable against the ego-motion of the camera, only the 3D version is able to re-track the target after it was hidden.

(c) Due to the 3D version using also depth information to identify occlusions, it manages to handle this scenario in contrary to the 2D version.
An additional flaw is the assumption of the target being able to move on the ground without constraints. Whereas, in the real world, the tracked objects i.e. vehicles are bounded to move on the road, which delimits their possible positions as shown in figure 18c. The 3D version assumes that the target is following a linear path, in accordance with the last estimated position and velocity where the target was still visible.

(a) A distractor being considered as the target after the actual target was hidden and recognized as occluded.

(b) The vehicle slows down and the state estimator is no able to catch up with the new speed, due to the constant velocity model adopted for the transition model of the state estimator.

(c) A case where the 3D version follows the prediction, but does not take into account the constraints of the real world, i.e the delimitation of the road.

Figure 18: Occlusion scenarios where the 3D version of DiMP was not able to handle occlusion correctly.

Naturally, there are cases where all trackers have failed as illustrated in figure 19 where their appearance model might be too similar to the car rather than the bicycle.
6 Conclusion

In this paper, we propose an approach to improve UAV on-board visual tracking. The approach combines information extracted from a visual tracker with a 3D reconstruction of the static environment observed by the tracker. Through the 3D reconstruction, we are able to use a state estimator in a 3D space rather than in the image space. Thus, allowing the state estimator to utilize a more realistic motion model that is more faithful to reality.

The potential of the approach is shown on prototypical data, reflecting typical real-world occlusion situations, captured from a low-altitude UAV. The experiments demonstrate that the presented approach is viable for effectively handling occlusions, low object-to-image ratio, and to provide stable tracking against ego-motion.

A part of our future work will be to exploit a dense reconstruction rather than a sparse reconstruction, explore different state estimators, and add more context to the scene e.g. by adding the layout of the road in the reconstruction.

Acknowledgments. We would like to thank Ann-Kristin Grosselfinger for providing the visual tracking ground truth data of the AU-AIR-Track dataset.

References

[1] Black, J., Ellis, T., Rosin, P., et al., “A novel method for video tracking performance evaluation,” Proceedings of the IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS 03), 125–132 (2003).

[2] Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Pflugfelder, R., Kamarainen, J.-K., Cehovin Zajc, L., Drbohlav, O., Lukezic, A., Berg, A., et al., “The seventh visual object tracking vot2019 challenge results,” in Proceedings of the IEEE International Conference on Computer Vision Workshops, 0–0 (2019).
[3] Wu, Y., Lim, J., and Yang, M.-H., “Object tracking benchmark,” IEEE Transactions on Pattern Analysis and Machine Intelligence 37(9), 1834–1848 (2015).

[4] Bozcan, I. and Kayacan, E., “Au-air: A multi-modal unmanned aerial vehicle dataset for low altitude traffic surveillance,” arXiv preprint arXiv:2001.11737 (2020).

[5] Danelljan, M., Bhat, G., Khan, F. S., and Felsberg, M., “Atom: Accurate tracking by overlap maximization,” in [The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)], (June 2019).

[6] Bhat, G., Danelljan, M., Gool, L. V., and Timofte, R., “Learning discriminative model prediction for tracking,” in [Proceedings of the IEEE International Conference on Computer Vision], 6182–6191 (2019).

[7] Schönberger, J. L. and Frahm, J.-M., “Structure-from-motion revisited,” in [Conference on Computer Vision and Pattern Recognition (CVPR)], (2016).

[8] Schönberger, J. L., Zheng, E., Pollefeys, M., and Frahm, J.-M., “Pixelwise view selection for unstructured multi-view stereo,” in [European Conference on Computer Vision (ECCV)], (2016).

[9] Schönberger, J. L., Price, T., Sattler, T., Frahm, J.-M., and Pollefeys, M., “A vote-and-verify strategy for fast spatial verification in image retrieval,” in [Asian Conference on Computer Vision (ACCV)], (2016).

[10] Wu, Y., Lim, J., and Yang, M.-H., “Online object tracking: A benchmark,” in [IEEE Conference on Computer Vision and Pattern Recognition (CVPR)], (2013).

[11] Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pfugfelder, R., Cehovin Zajc, L., Vojir, T., Bhat, G., Lukezic, A., Eldesokey, A., et al., “The sixth visual object tracking vot2018 challenge results,” in [Proceedings of the European Conference on Computer Vision (ECCV)], 0–0 (2018).

[12] Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., and Ling, H., “Lasot: A high-quality benchmark for large-scale single object tracking,” in [The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)], (June 2019).

[13] Smeulders, A. W., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., and Shah, M., “Visual tracking: An experimental survey,” IEEE transactions on pattern analysis and machine intelligence 36(7), 1442–1468 (2013).

[14] Valmadre, J., Bertinetto, L., Henriques, J. F., Tao, R., Vedaldi, A., Smeulders, A. W., Torr, P. H., and Gavves, E., “Long-term tracking in the wild: a benchmark,” in [The European Conference on Computer Vision (ECCV)], (September 2018).
[15] Huang, L., Zhao, X., and Huang, K., “Got-10k: A large high-diversity benchmark for generic object tracking in the wild,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 1–1 (2019).

[16] Liang, P., Blasch, E., and Ling, H., “Encoding color information for visual tracking: Algorithms and benchmark,” IEEE Transactions on Image Processing 24(12), 5630–5644 (2015).

[17] Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., and Ghanem, B., “Trackingnet: A large-scale dataset and benchmark for object tracking in the wild,” in [The European Conference on Computer Vision (ECCV)], (September 2018).

[18] Li, A., Chen, Z., and Wang, Y., “Buaa-pro: A tracking dataset with pixel-level annotation,” in [BMVC], 249 (2018).

[19] Bolme, D. S., Beveridge, J. R., Draper, B. A., and Lui, Y. M., “Visual object tracking using adaptive correlation filters,” in [2010 IEEE computer society conference on computer vision and pattern recognition], 2544–2550, IEEE (2010).

[20] Danelljan, M., Hager, G., Shahbaz Khan, F., and Felsberg, M., “Learning spatially regularized correlation filters for visual tracking,” in [Proceedings of the IEEE international conference on computer vision], 4310–4318 (2015).

[21] Danelljan, M., Robinson, A., Khan, F. S., and Felsberg, M., “Beyond correlation filters: Learning continuous convolution operators for visual tracking,” in [European conference on computer vision], 472–488, Springer (2016).

[22] Kiani Galoogahi, H., Sim, T., and Lucey, S., “Correlation filters with limited boundaries,” in [Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition], 4630–4638 (2015).

[23] Henriques, J. F., Caseiro, R., Martins, P., and Batista, J., “High-speed tracking with kernelized correlation filters,” IEEE transactions on pattern analysis and machine intelligence 37(3), 583–596 (2014).

[24] Ma, C., Huang, J.-B., Yang, X., and Yang, M.-H., “Hierarchical convolutional features for visual tracking,” in [Proceedings of the IEEE international conference on computer vision], 3074–3082 (2015).

[25] Sun, C., Wang, D., Lu, H., and Yang, M.-H., “Correlation tracking via joint discrimination and reliability learning,” in [Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition], 489–497 (2018).

[26] Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., and Torr, P. H., “Fully-convolutional siamese networks for object tracking,” in [European conference on computer vision], 850–865, Springer (2016).
[27] Guo, Q., Feng, W., Zhou, C., Huang, R., Wan, L., and Wang, S., “Learning dynamic siamese network for visual object tracking,” in [Proceedings of the IEEE international conference on computer vision], 1763–1771 (2017).

[28] Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., and Yan, J., “Siamrpn++: Evolution of siamese visual tracking with very deep networks,” in [Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition], 4282–4291 (2019).

[29] Li, B., Yan, J., Wu, W., Zhu, Z., and Hu, X., “High performance visual tracking with siamese region proposal network,” in [Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition], 8971–8980 (2018).

[30] Wang, Q., Teng, Z., Xing, J., Gao, J., Hu, W., and Maybank, S., “Learning attentions: residual attentional siamese network for high performance online visual tracking,” in [Proceedings of the IEEE conference on computer vision and pattern recognition], 4854–4863 (2018).

[31] Zhang, L., Gonzalez-Garcia, A., Weijer, J. v. d., Danelljan, M., and Khan, F. S., “Learning the model update for siamese trackers,” in [Proceedings of the IEEE International Conference on Computer Vision], 4010–4019 (2019).

[32] Mueller, M., Smith, N., and Ghanem, B., “A benchmark and simulator for uav tracking,” in [European conference on computer vision], 445–461, Springer (2016).

[33] Li, S. and Yeung, D.-Y., “Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models,” in [AAAI], (2017).

[34] Du, D., Qi, Y., Yu, H., Yang, Y., Duan, K., Li, G., Zhang, W., Huang, Q., and Tian, Q., “The unmanned aerial vehicle benchmark: Object detection and tracking,” in [Proceedings of the European Conference on Computer Vision (ECCV)], 370–386 (2018).

[35] Zhu, P., Wen, L., Du, D., Bian, X., Hu, Q., and Ling, H., “Vision meets drones: Past, present and future,” arXiv preprint arXiv:2001.06303 (2020).

[36] Osep, A., Mehner, W., Mathias, M., and Leibe, B., “Combined image- and world-space tracking in traffic scenes,” in [ICRA], (2017).

[37] Geiger, A., Lenz, P., and Urtasun, R., “Are we ready for autonomous driving? the kitti vision benchmark suite,” in [Conference on Computer Vision and Pattern Recognition (CVPR)], (2012).

[38] Menze, M. and Geiger, A., “Object scene flow for autonomous vehicles,” in [Conference on Computer Vision and Pattern Recognition (CVPR)], (2015).
[39] Luiten, J., Fischer, T., and Leibe, B., “Track to reconstruct and reconstruct to track,” *IEEE Robotics and Automation Letters* **5**, 1803–1810 (2020).

[40] He, K., Zhang, X., Ren, S., and Sun, J., “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778 (2016).

[41] Jiang, B., Luo, R., Mao, J., Xiao, T., and Jiang, Y., “Acquisition of localization confidence for accurate object detection,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 784–799 (2018).

[42] Douc, R. and Cappé, O., “Comparison of resampling schemes for particle filtering,” in *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, 64–69, IEEE (2005).

[43] Lukezic, A., Zajc, L. C., Vojir, T., Matas, J., and Kristan, M., “Now you see me: evaluating performance in long-term visual tracking,” *arXiv preprint arXiv:1804.07056* **4** (2018).