DronePose: Photorealistic UAV-Assistant Dataset Synthesis for 3D Pose Estimation via a Smooth Silhouette Loss

Georgios Albanis[*1], Nikolaos Zioulis[*1], Anastasios Dimou[1], Dimitrios Zarpalas[1], and Petros Daras[1]

Visual Computing Lab, Information Technologies Institute, Centre for Research and Technology Hellas, Greece
{galbanis, nzioulis, dimou, zarpalas, daras}@iti.gr
vcl.iti.gr

Abstract. In this work we consider UAVs as cooperative agents supporting human users in their operations. In this context, the 3D localisation of the UAV assistant is an important task that can facilitate the exchange of spatial information between the user and the UAV. To address this in a data-driven manner, we design a data synthesis pipeline to create a realistic multimodal dataset that includes both the exocentric user view, and the egocentric UAV view. We then exploit the joint availability of photorealistic and synthesized inputs to train a single-shot monocular pose estimation model. During training we leverage differentiable rendering to supplement a state-of-the-art direct regression objective with a novel smooth silhouette loss. Our results demonstrate its qualitative and quantitative performance gains over traditional silhouette objectives. Our data and code are available at https://vcl3d.github.io/DronePose.

Keywords: 3D Pose Estimation, Dataset Generation, UAV, Differentiable Rendering

1 Introduction

Advances in robotics, their autonomy, and computer vision are constituting Unmanned Aerial Vehicles (UAVs) – or otherwise known as drones – an emerging ubiquitous technology. Indeed, the potential that autonomous or piloted UAVs can provide has already been acknowledged in numerous application fields like filming [65], emergency response [31], animal conservation [27], and infrastructure inspection [14]. In addition, a lot of research has focused on the detection of drones in the context of anti-drone systems [55] and their accurate identification [10].

Nevertheless, the miniaturization and commoditisation of drones open up new types of applications for mini-UAVs. Moving beyond the role of remote, and/or malicious agents, personal – friendly – drones can operate in tandem

---

[*] Equal contribution

with human users, supporting their activities and acting as external assistants [3]. Yet, human operators can also be augmented with modern sensing capabilities (*e.g.* HoloLens), enabling advanced cooperation schemes like XRay vision [13]. Under such conditions, the registration of the human operator and the peer drone is an important task as it facilitates the exchange of spatial information, enabling functionalities like gaze-driven navigation [66].

To address friendly UAV 3D pose estimation in a data-driven manner, we first need to overcome the lack of data. All UAV related datasets either target remote sensing applications, and therefore, offering purely drone egocentric views, or consider drones as malicious agents and offer (multi-modal) sensor exocentric views. However, datasets created for anti-UAV applications cannot be re-used in UAV-Assistant (UAVA) settings mainly because of the very large distance between the sensors and the drones. This is not only out of context, but also suffers from low-resolution modalities (*i.e.* thermal), tiny object image sizes, and a serious lack of finer-grained discriminating information for 3D pose estimation.

The availability of UAV simulators integrated with high-end rendering inside game engines, like AirSim [54] and Sim4CV [39], offers a promising direction for synthesizing UAVA data. This synthetic dataset generation approach has been used for spacecraft pose estimation [47] and generating low-altitude flight data [27]. Still, the synthetic to real domain gap will hinder performance when applied to actual real-world cases. While approaches like FlightGoggles [17] mitigate this gap through photorealistic sensor simulation, actual real-world data are still preferable. Towards that end, we follow a principled approach to create a realistic multimodal UAVA dataset. We employ real-world 3D scanned datasets, physically-based shading, a gamified simulator for realistic drone navigation trajectory collection and randomized sampling, to generate multimodal data both from the user's exocentric view of the drone, as well as the drone's egocentric view. This allows us to train a CNN for single-shot drone 3D pose estimation from monocular (exocentric) input. Given the availability of projected drone silhouettes aligned with the exocentric view, we complement our model with an enhanced smooth silhouette consistency loss. Overall, our contributions are the following:

- We create a realistic multimodal UAVA dataset that offers aligned exocentric (user) and egocentric (drone) data along with the ground-truth poses. We also consider an inter-frame motion for both views, with the ego-motion supported by the scene's optical flow, while the exocentric view provides the drone's optical flow.

- We develop an end-to-end model that combines direct pose regression and differentiable rendering for learning 6DOF pose estimation and train it with complex color images to explore their combined potential.

- We propose a smooth silhouette loss that improves performance compared to traditional alternatives as well as model robustness.

2   Related Work

UAV Datasets: Most existing UAV datasets can be categorized into two main categories, datasets for remote sensing and anti-UAV datasets. The former utilize drones as remote eyes observing remote scenes and aim at either applying real-time computer vision algorithms on live data streams, or recording them for offline analysis. The VisDrone2019 dataset [70] contains diverse data captured from drone-mounted cameras and is annotated with bounding boxes for object detection and tracking tasks, focused on the surveillance aspect of UAVs. Similar data is included in the Stanford Drone Dataset [51] with various bounding box annotations of different classes found in a campus, with the distinct difference being its strict top-down view. The VIRAT video surveillance dataset [40] also contains UAV captured scenes which, apart from object annotations, also includes activity and event annotations. Apart from these close-range egocentric view datasets, typical remote sensing drone datasets involve ground observing aerial images. Examples include the urban settlement dataset of [35], or the Kuzikus dataset used in [27] which contains wildlife detection annotations. Similar aerial top-down view datasets are used for object counting [22] and tracking [38]. Summarizing, datasets that view UAVs as remote sensors, offer the drone's egocentric view, usually with object (*i.e.* bounding box) or image segment (*i.e.* mask) annotations.

The second UAV dataset category, considers UAVs as hostile agents, a topic that is gaining significant traction with the research community lately, mainly because of the challenges it entails. Anti-UAV systems need to operate in-the-wild, in adverse weather conditions, during the whole day-night cycle, and offer timely and accurate information for any neutralizing actions to take place. Towards that end, the available datasets address tiny object detection [52] and adversarial conditions that involve birds [10]. Recent organized challenges like the Anti-UAV challenge [1] include multimodal (*i.e.* color and thermal) inputs to increase detection robustness under different lighting and weather conditions.

All the aforementioned datasets are captured with traditional sensors and are manually annotated. Nonetheless, recent advances in UAV simulators, a prominent one being AirSim [54], enable the generation of data via synthesis, and thus, the availability of information that is traditionally very hard to acquire with traditional means (*i.e* depth, surface orientation, semantics, 3D pose, etc.). This was the case of the AimSim-w dataset [4] which synthesized remote sensing data for wildlife observation, in addition to modelling the thermal imaging process to synthesise infrared data. Similarly, the Mid-Air dataset [15] used AirSim to synthesize low-altitude flight multimodal data and illustrated its efficacy in drone depth estimation.

Still, the domain gap is an important obstacle in using fully synthetic datasets that employ purely computer-generated models and imagery. While domain adaptation itself is another important task, with the reader referred to a survey [62] for more details, we can also find other intermediary approaches in the literature. FlightGoggles [17] is a photorealistic sensor simulator that infuses realism in a data generation pipeline for UAVs. It was used when creating the

aggressive drone flights Blackbird dataset [2], which does not fit in the two afore-mentioned categories in the broader sense, as it is used for visual-inertial SLAM benchmarking. Still, the synthesis of realistic data, as encompassed by Flight-Goggles and the Blackbird dataset is highly related to our approach. We also use scanned 3D models like the photogrammetry models used in FlightGoggles and generate external trajectories like those used the Blackbird dataset. Contrary to those though, we collect UAV flight trajectories with significantly higher variance than the periodic Blackbird ones, and utilize a vastly larger 3D scene corpus than FlightGoggles. Moreover, as it is apparent by the above dataset analysis (with more details available in the supplement), none are suitable for UAVA applications as the exocentric anti-UAV datasets image drones from very far distances, similar to how egocentric remote sensing drone datasets image (very) far scenes. In the UAVA setting, the user and the drone work under a cooperative context that allows for the availability of both the egocentric and exocentric views simultaneously, and will typically be close in close proximity.

Data-driven 6DOF Pose Estimation: Even before the establishment of deep models, data-driven methods relying on random forests [12,57,5] or deformable part models [45] started producing high-quality results for the task of 3D pose estimation. Early deep models [37,18,56,53] focused on learning rotations (*i.e.* viewpoints) and performed classification after binning them, using either normal maps [18], synthesized [56] or traditional [37] images, or even employed colorized depth inputs [53]. Follow-up works transitioned from binned rotation classification to direct regression of the 6DOF camera pose using varying rotation representations. PoseCNN [64] used a quaternion representation and employed learned segmentation predictions to handle cluttered object scenes, while [36] demonstrated results for an axis-angle representation in addition to the quaternion one. Currently, the state-of-the-art has been focusing on addressing the discontinuity of rotations by disentangling the rotation manifold [33] or exploring continuous rotation representations [69]. Single-shot variants predict the 3D bounding boxes from monocular inputs [58], or formulate multiple pose hypotheses from 2D bounding box detections [26], solving either a Perspective-n-Point (PnP) problem or locally optimizing the pose afterwards respectively. A slightly different approach has been introduced in [48] where the image is cropped around the object by first predicting a segmentation mask and then the 2D bounding box is predicted. While most works are trained on real-world images, in [49] it was shown that using synthetic images and single channel edge information can lead to high-quality results, even when directly regressing the 6DOF pose. Even though direct pose regression has demonstrated high-quality results, especially under iterative frameworks [60], most recent research has resorted to correspondence-based pose estimation. Deep keypoint prediction was originally used for 3D pose estimation in [59] and has since spun to a variety of approaches even in drone pose estimation[25]. PVNet [44] votes for keypoints, which are then used to estimate the object's 6DOF pose using PnP. Following a dense correspondence approach, various works directly regress either 2D image to 3D model correspondences [67], normalized object coordinates [61], or leverage

deep Hough voting as in the case of PVN3D [20]. Regarding coordinates based regression, separate branches for the translation and rotational components were found to be more robust in predict pose [32], especially for occluded and texture-less objects. More recently, the various advances in differentiable rendering are enabling end-to-end 3D vision, and naturally, their application to 6DOF pose estimation is being explored. A differentiable point cloud projection function was employed [23] that used two simplistic views (*i.e.* white background, silhouette-like images) of an object to jointly estimate its shape and pose, using an MSE silhouette loss. Similarly, a rendering-like occlusion removal operator was added in [63] to handle point cloud viewpoint-based self-occlusions and drive supervision solely from depth images. Given that only the pose needs to be optimized, a lightweight abstract rendering mechanism based on OpenDR [34] was developed in [46] to compare learned feature representations from meshes and the observed images, and then estimate their poses in the observed scenes. LatentFusion [42] learns a reconstructed feature representation that can be rendered in order to optimize for an object's pose as it is observed within an image, and has been shown to also generalize well to unseen objects. Finally, a rendered silhouette error is backpropagated in [41] to refine the CNN estimated pose. However, it was only applied in simplistic CAD renderings and was not trained in an end-to-end fashion. This was also the case for [23] which did not use monocular input and only learned pose estimation using simplistic images. Instead, we leverage end-to-end differentiable rendering using complex imagery for boosting 6DOF pose estimation.

3   Dataset

Our goal is to create a realistic UAVA dataset for applications in which users and drones are considered cooperative agents. To achieve this, we need to overcome a set of challenges associated with the photorealism of the visual data, the plausibility of the drone-to-user spatial relation, the variety/plurality of the generated views, and the drone's trajectories. Contrary to similarly oriented datasets like URSO [47] and Blackbird [2] we are not confined to only a single ego- [2] or exocentric view [47], and thus, we need to consider both viewpoints and synthesize temporally and spatially aligned data. While both [47] and [2] render purely computer-generated scenes, we rely on a recent 3D scanned dataset, Matterport3D [7] that offers us scene photorealism.

The Matterport3D dataset was created after capturing multiple RGB-D panoramas within large buildings. As the poses of each panorama are available, we can leverage them as stationary points (*i.e.* anchors) to drive our UAVA oriented data generation. Aiming to create realistic drone flight trajectories and simultaneously position user viewpoints in a plausible way, we employ a gamified approach. More specifically, we designed and developed a Unity3D game where collectible cube "coins" are placed at each of the known panorama positions (anchors) and use AirSim's drone flight simulator to let players navigate a drone within each building to collect the pre-placed "coins" using a game controller.
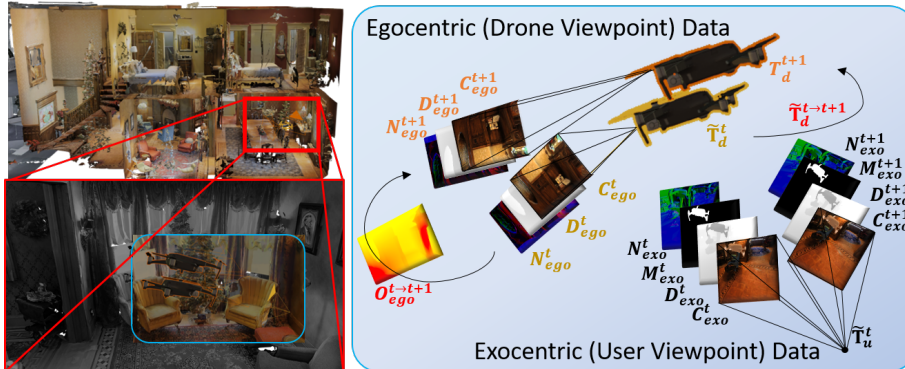
Fig. 1. We use a large-scale photorealistic 3D scene dataset for rendering our UAVA dataset. After sampling a set of drone navigation trajectories created via a gamification process, we raycast data from a drone egocentric view as well as a user's exocentric one. For the former, we generate color (C) images, depth (D) and surface (N) maps, in addition to the optical flow (OF) for two consecutive frames $t, t + 1$ sampled from the dense play-through trajectories. Apart from the ego-motion data, the exocentric viewpoints image the drone as posed in the $t, t+1$ frames and offer the same modalities, in addition to the drone's optical flow in the statically observed scene, and its silhouette images (M). Supplementing the images, our dataset offers precise pose annotations for the drone and the user viewpoints.

The players need to collect all "coins" placed inside the building as soon as possible, while at the same time a leaderboard records the quickest runs for boosting engagement among users and adding trajectory diversity. Each run is initialised from a randomized anchor location to further increase trajectory diversity (see Fig. 2). The game controller together with the simulation framework allows for quicker and more intuitive drone navigation. This, combined with expanded colliders for the "coins", the initialization randomness and the timing objective adds the necessary movement pattern variety in our collected trajectories.

The drone's world pose $\mathbf{T}_d^t$ is recorded at each time step t and can be associated with an anchor point a. The drone's pose to anchor's pose association is done on a per nearest Euclidean neighbor basis, thereby providing us with a pair $P := (\mathbf{T}_a, \mathbf{T}_d^t)$ at each time step. As each game run's $60Hz$ trajectories are recorded, we need to sample them to reduce redundancy, filter potentially bad samples due to occlusions and 3D model holes, and also add additional randomness to increase our dataset's variance as follows.

We sample sparse pose samples from the dense trajectories as follows. We first filter all pairs whose Euclidean distance is over $1.5m$ and under $0.8m$, effectively limiting the user-to-drone distance to this range. We then apply ego-motion criteria to the remaining pairs across time, filtering out poses that contain limited positional and angular displacement with respect to their inter-frame poses at $t - 1$ and $t + 1$. Further, we filter out poses that do not include the entirety of

Fig. 2. Our gamified approach for generating high-quality diversified data. The first column presents the top-view of a Matterport3D building overlayed with the trajectory generated by a single play-through. The second column presents cuboid "coins" placed at the anchor points within the game. These collectible coins ensure that the players will cover all of the building's area and will traverse it near the anchor points. The third column collates two side views of the same building while the last column offers two in-game screenshots.

the drone by removing those where the drone's centroid is projected near the image boundaries, and we also remove samples that contain a lot of invalid pixels (*i.e.* large mesh holes) via percentage-based thresholding. For the remaining pose samples, we introduce a "look at" variance by uniformly sampling an axis-aligned bounding cube around each anchor point $\mathbf{T}_a$ and the drone's pose $\mathbf{T}_d^t$ and setting the exocentric view's origin at the anchor cube sampled point, looking at the drone cube sampled point, generating a randomized user $\mathbf{T}_u^t$.

We then render ego-motion data after pairwise temporally grouping the remaining pairs and using the drone's poses $(\mathbf{T}_d^t, \mathbf{T}_d^{t+1})$. We output color, depth, surface, and the optical flow between them using raytracing. For the exocentric renders we employ a CAD 3D drone model, enhanced with a physically based rendering (PBR) material with a bidirectional scattering distribution function (BSDF) to increase the realism of its renders. We place the drone at each pose pair $(\mathbf{T}_d^t, \mathbf{T}_d^{t+1})$ and then render the scene from each user $\mathbf{T}_u^t$ twice. Essentially, we use the dataset's anchor points as user views, after forcing the drone to run by them in various directions. For the exocentric user views, we output color, depth, surface and the drone's silhouette, in addition to its optical flow between the successive poses $(\mathbf{T}_d^t, \mathbf{T}_d^{t+1})$. An illustration of this process can be found in Fig. 1. Another important factor that should be considered while creating synthetic data is lighting. A recent study has identified the importance of the illumination model, as well as that a proper modelled lighting environment can give the same results as a natural environmental light [68].

Additionally, given that Matterport3D's scanned models contain pre-baked lighting and the drone model is unlit, a straightforward rendering with no light, or the addition of random lighting would produce unrealistic results as in the
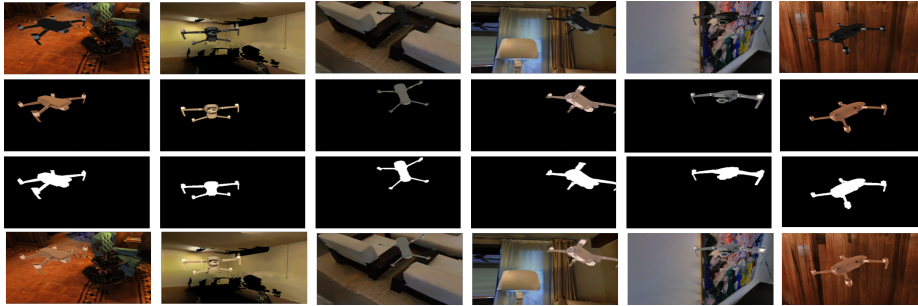
Fig. 3. Our compositing technique for producing realistic results. The top row illustrates the unlit drone model rendered in the already lit scene which produces a consistent appearance across all samples. The second row presents the same model rendered with our randomized lighting, while the third row depicts its anti-aliased silhouette. Finally, the bottom row presents the final composited result where the lit drone has been blended naturally in the environment.

former case, the drone's appearance would be very consistent and dark across all renders, while in the latter case the already lit environment would become more saturated.

Therefore, we resort to a compositing approach where the scene is rendered twice, once with no lighting (preserving the emission of the already lit environment) and then, a follow up render which only renders the drone with random lighting as well as its mask. We use an advanced matting technique [16] to perform high-quality compositing of the lit drone to the unlit render with almost no aliasing using the drone mask, as presented in Fig. 3. Taking into account the findings of [68], for the drone lighting, we use a random texture of the rendered scene as an environment map with random light strength drawn from a uniform distribution $l \in \mathcal{U}(0.3, 0.7)$. This offers natural lighting for the 3D model that gets composited into the pre-baked lit environment, while also preserving a certain level of randomness, and thus variety.

Fig. 4 presents the final dataset distribution of the train and test splits in terms of the $x - z$ and $y - z$ 3D coordinates density in the user's coordinate system, as well as the drone's yaw, pitch, roll rotations. It can be seen that our gamified and sampling-based approach has led to a smooth distribution of poses around the local exocentric viewpoint, with consistency between the train and test splits. In addition, a large enough variance of plausible drone rotations ensures a high-quality dataset for learning and benchmarking 6DOF pose estimation.

4   DronePose

In this section, we describe our single-shot approach in learning to estimate the 6DOF pose of a known UAV using our generated dataset. We address this purely
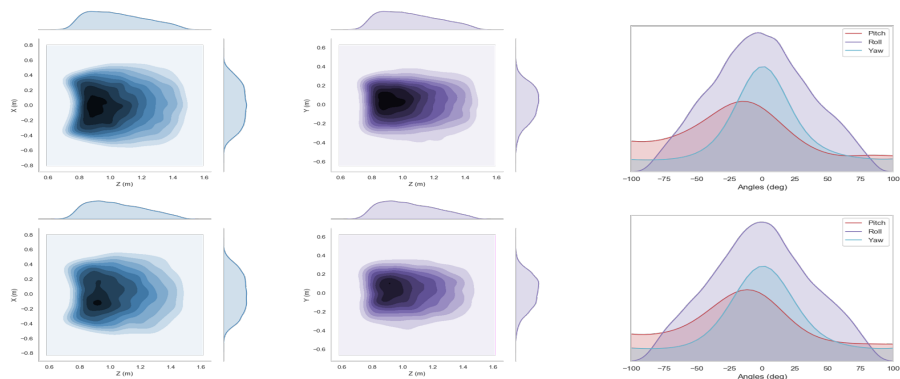
Fig. 4. This figure presents the drone pose distribution on the local coordinate system of the exocentric viewpoint. The top row presents the train split distribution, while the bottom row presents the test split distribution. From left to right: **i)** the x axis 3D coordinate distribution with respect to the depth (z), **ii)** the same for the y axis, and **iii)** the drone's yaw, pitch and roll angle distributions.

from monocular input from the exocentric view in a supervised manner. Given the user's exocentric pose $\mathbf{T}_u^t$ and the drone's pose $\mathbf{T}_d^t$ in the world coordinate system, we can extract their relative groundtruth pose $\mathbf{T} = \mathbf{T}_u^{-1}\mathbf{T}_d := \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$, omitting the time index for brevity for the remainder of this document. Inspired by [32] we use a backbone CNN encoder, followed by three fully connected linear units which then disentangle into two prediction heads for the translation and rotation components (see Fig. 5). We employ a state-of-the-art continuous $6d$ rotation parameterisation [69] that is transformed to a $9d$, 3×3 rotation matrix $\tilde{\mathbf{R}}$[1], which when combined with the $3d$ translation head's vector output $\tilde{\mathbf{t}}$, forms a homogeneous 4×4 pose $\tilde{\mathbf{T}} := \begin{bmatrix} \tilde{\mathbf{R}} & \tilde{\mathbf{t}} \\ \mathbf{0} & 1 \end{bmatrix}$. We use a L2 loss for the translation $\mathcal{L}_t = \|\mathbf{t} - \tilde{\mathbf{t}}\|_2^2$, while for the rotation we minimize their inner product on $SO(3)$, or otherwise their angular difference $\mathcal{L}_r = \arccos \frac{trace(\mathbf{R}\tilde{\mathbf{R}}^T) - 1}{2}$. The final direct pose regression objective is:

$$\mathcal{L}_{pose} = \lambda_{pose}\,\mathcal{L}_t + (1 - \lambda_{pose})\,\mathcal{L}_r\,, \tag{1}$$

with the ratio λ_{pose} balancing the errors between the rotation and translation.

Aiming to estimate the pose of a known object whose 3D model is available, and given the availability of high-quality silhouette masks, we support the direct regression loss with a rendered silhouette loss by employing a differentiable renderer. This is also an advantage of our chosen $6d$ representation [69] which in contrast to other state-of-the-art representations like [33,26,48,58], allows for the end-to-end integration of a rendering module. The approach of [33] splits the quaternion's components sign values into a separate classification branch, a technique that prevents its differentiable reformulation into the actual rotation

---

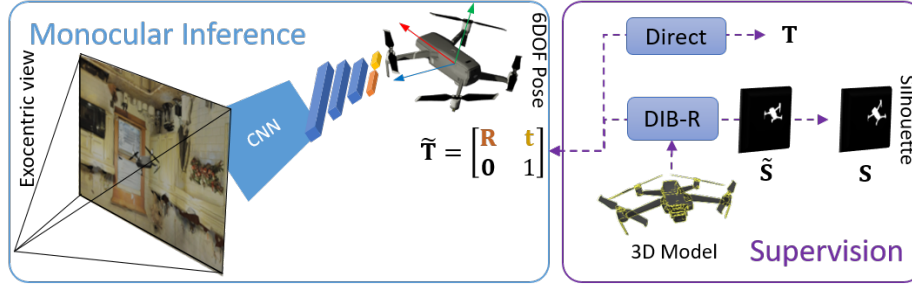[1] The accent denotes model predicted values.

Fig. 5. We introduce a dataset for UAV-Assistant applications that consider UAVs as friendly cooperative agents. We exploit a part of this dataset for developing a monocular (from exocentric views), UAV 6DOF pose estimation method. We enhance a state-of-the-art direct pose regression model with a differentiable rendering (i.e. DIB-R) silhouette consistency objective to improve performance, leveraging our fully differentiable approach.

and can only support pure direct pose regression models. Similarly, for dense keypoint voting or correspondence representations, a soft differentiable approximation would be required to turn this intermediate representation into a rotation matrix.

We transform the drone 3D model's vertices $\mathcal{V} \in \mathbb{R}^3$ using the predicted pose $\tilde{\mathbf{T}}^2$, and then use the recently introduced DIB-R differentiable mesh renderer [8] to render the transformed model's silhouette as observed by the user's viewpoint. Our final loss function supplements direct regression with an exocentric objective

$$\mathcal{L} = (1 - \lambda_{exo}) \, \mathcal{L}_{pose} + \lambda_{exo} \, \mathcal{L}_{exo}, \tag{2}$$

with λ_{exo} balancing the contribution of the exocentric loss ($\mathcal{L}_{exo}$). For the latter, a per pixel silhouette consistency loss function, formulated either as an L2 or binary cross entropy loss, would be a sub-optimal choice as it is an asymmetric objective. Indeed, through a differentiable renderer, only the pixels of the predicted silhouette would back-propagate error, neglecting any errors associated with the groundtruth silhouette proximity. This, in combination with the constant error in the non-overlapping cases, contribute to a loss surface that is constant, suffers from local minima, and cannot provide meaningful gradient flows. A typical choice in the literature is the Jaccard [30] loss, or otherwise referred to as the intersection-over-union (IoU) metric, which as a loss is defined as:

$$\mathcal{L}_{iou} = \frac{1}{N} \sum_{\mathbf{p} \in \Omega} 1 - \frac{\mathbf{S} \odot \tilde{\mathbf{S}}}{\mathbf{S} \oplus \tilde{\mathbf{S}} - \mathbf{S} \odot \tilde{\mathbf{S}} + \epsilon}, \tag{3}$$

with $\mathbf{S}$ and $\tilde{\mathbf{S}}$ being the binary groundtruth and rendered predicted silhouette images respectively, $\mathbf{p}$ corresponding to a pixel in the image domain Ω that the

---

[2] 4D homogenisation of the 3D coordinates is omitted for brevity.

silhouettes are defined in, N being the total image domain elements, and ϵ a small numerical stabilization constant.

However, a known weakness of the Jaccard loss (which is a similarity measure metric by mathematical definition) is its plateau when there is no overlap between the silhouettes, offering no notion of proximity to the objective itself. In addition, taking into account that our loss is defined on the projective space, the reduced degrees-of-freedom manifest into loss irregularities, especially due to rotations and the 3D shape variations. Motivated by these shortcomings, we propose to use a smooth silhouette loss defined as:

$$\mathcal{L}_{smooth} = \frac{1}{N} \sum_{\mathbf{p} \in \Omega} \mathbf{S} \odot \mathcal{S}(\tilde{\mathbf{S}}) + \tilde{\mathbf{S}} \odot \mathcal{S}(\mathbf{S}), \qquad (4)$$

where the function $\mathcal{S}$ calculates a truncated, continuous silhouette proximity map $\mathcal{S}(\mathbf{S}) = (g \circledast \overline{\mathbf{S}}) - (g \circledast \mathbf{S})$, where $\overline{\mathbf{S}} := 1 - \mathbf{S}$ is the complement of silhouette $\mathbf{S}$ and g is a convolved smoothing kernel. Our smooth silhouette loss exhibits some very important traits, it is quick to evaluate as it is fully parallelizable and it offers a smooth objective function compared to IoU, enhanced with a notion of proximity to the objective. This is controlled by the convolution's kernel size which also adjusts the smoothed silhouette region against the truncated one. Finally, it is a fully symmetric objective that takes into account the groundtruth silhouette as well. More importantly, this is achieved in a fully differentiable manner with respect to the predicted silhouette. Fig. 6 illustrates the difference between IoU (Eq. 3) and our introduced smooth loss (Eq. 4) across the silhouettes produced by a dense sampling of poses around a target pose and its silhouette, showcasing the smoothness and advantages of our objective function[3].

5 Results

In this section, we explore the supplementary nature of the exocentric loss for the task of object 3D pose estimation through a set of structured experiments.

Implementation Details: When generating our dataset, we follow the official Matterport3D data scene splits, and thus, our final rendered dataset comprises 38155 train samples, 5687 validation samples and 11146 test samples. We used PyTorch [43] and Kaolin [24] for the implementation of our model. In more detail, we employ a ResNet34 [19] pre-trained on ImageNet [11] as our backbone encoder, followed by fully connected layers for reducing the encoded features progressively by using $1024, 512$ and 128 features respectively, which eventually branch off to the $6d$ and $3d$ rotation and translation heads. We use ELU [9] activation functions for the linear layers, as well as the Adam [28] optimizer with its default momentum parameters. For the smoothing kernel we present results for an efficient 49×49 low-pass box smoothing filter, as well as a 69×69

---

[3] Additional analysis can be found in our supplement including a loss landscape analysis.
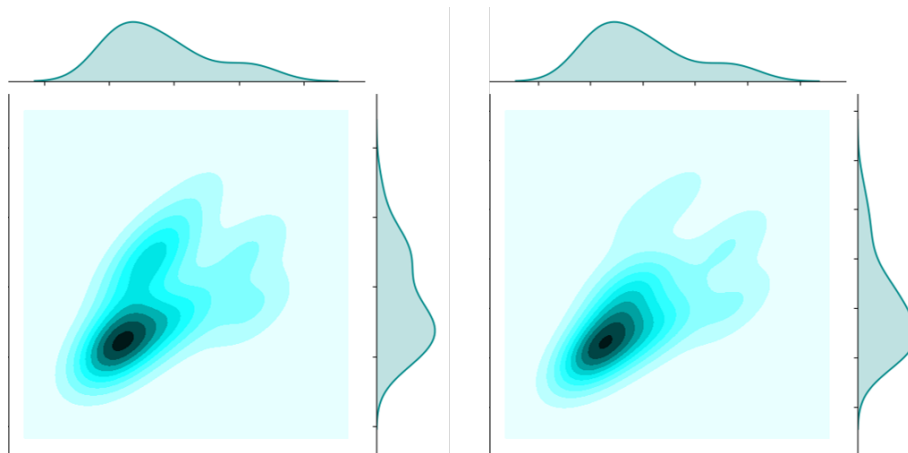
Fig. 6. Distribution of the IoU (left) and our smooth silhouette (right) losses across a dense sampling of poses. Due to the inherent difficulty in plotting 6DOF poses, the presented distributions are generated with the following process. We perform a grid search of 3DOF translational and 3DOF angular displacements, and then bin them according to their Euclidean and angular distances respectively. The binning process accumulates the error of the silhouette rendered by displaced pose against the original pose. Given the different scale of the two loss functions, we first apply min-max normalization before binning and accumulating them. The plots then present the distribution of these bins, with the translational radius distance binned and accumulated errors on the horizontal axis, and the rotational angular distance binned and accumulated errors on the vertical axis. Our smoother loss variant produces a smoother curve for the rotational error as evident in the vertical axis. In addition, the larger number of transitions (*i.e.* color steps) and their smaller steps indicate an overall smoother objective. Further, a smaller convergence region around the groundtruth pose, as well as the smaller constant loss regions indicate that IoU suffers from harder convergence towards the minima.

Gaussian kernel. For the [58] we followed the official implementation. All models were trained for 20 epochs with a batch size of 64 and a learning rate of $1e-4$.

Metrics: We report the following evaluation metrics for all of our experiments, the normalized position error (NPE), the orientation error (OE) and the combined pose error (CPE) as defined in [29], as well as the 6D Pose error [21] and the combined position and orientation accuracy metric ($AccX$). For the latter, a pose is considered as accurate if it lies within an angular error under X degrees and a position error under Xcm. Similarly, the 6D Pose error is reported using the same thresholds as the accuracy metric, and we consider the 3D model's diagonal rather than its diameter.

Experiments: All results reported in Table 1 represent the best performing model across the total epochs. We demonstrate through extensive experiments how differential rendering can be employed for improving pose estimation, and also highlight the weaknesses of common loss functions (i.e. IoU, GIoU[50]).

Further, we compare our method with a single-shot variant [58] oriented towards real-time applications, which outperforms prior work [6,48,26]. Ablation results with respect to the smoothing kernel size and type can be found in our supplementary material. Our baseline is a *Direct* regression model using a state-of-the-art 6d representation [69], *i.e.* setting $\lambda_{exo} = 0.0$. Next, we enhance our model with an IoU-based supervision loss with $\lambda_{exo} = 0.1$ (*I0.1*) and $\lambda_{exo} = 0.2$ (*I0.2*). No apparent performance gain is observable for *I0.1*, while *I0.2* negatively affects the models performance. It should be noted that for higher λ_{exo} the model did not converge, indicating that the exocentric loss is not sufficient to drive optimization alone. Particularly, the IoU loss does not exhibit a smooth learning curve during training, a fact that hurts convergence due to the bigger effect it has while back-propagating gradients. Intuitively though, it should be more effective on a model that produces higher quality predictions as its plateau weakness would be circumvented. Therefore, our next set of experiments focus on increasing the IoU loss during training after a set of epochs. In more details, for the *I0.1-0.4* experiment, we increase the IoU weight λ_{exo} from 0.1 to 0.4 after 15 epochs and train for another 5 epochs. The *I0.1-0.3-0.4* experiment switches at epochs 8 and 12 at the respectively weight values, and the *I0.1-0.3-0.6* experiment accordingly switches at epochs 11 and 15. The results indicate that this is a valid strategy for improving the model's performance. Next, we present results for a GIoU [50] variant that we implemented oriented for silhouettes, using $\lambda_{exo} = 0.1$ (*G0.1*). While GIoU relies on non-differentiable *min* and *max* operations, it has demonstrated better performance than IoU for bounding box regression. In our case though, it does not demonstrate any performance gains even though it aims at reducing IoU's plateaus. The final experiments and results showcase the efficacy of our proposed smooth silhouette consistency loss and similar to the above experiments, are prefixed with an *S* followed by the λ_{exo} value. The best performing model is *Gauss0.1* that shows the largest gains compared to [58] and our baseline. At the same time, our box-filter implementation *S0.2,S0.2* and *S0.3* offer better performance as then IoU, and *S0.3* managed to

Table 1. Results on the generated dataset's test set which contains unseen images.

	NPE ↓	OE ↓	CPE ↓	Acc5 ↑	Acc10 ↑	6D Pose-5 ↑	6D Pose-10 ↑
Singleshotpose[58]	0.025	0.070	0.095	79.418	95.657	47.009	79.669
Direct	0.020	0.059	0.079	89.555	98.661	38.464	88.113
I0.1	0.020	0.059	0.079	88.880	98.661	42.348	87.782
I0.2	0.022	0.058	0.081	89.314	98.759	28.682	84.788
I0.1-0.4	0.020	**0.056**	**0.076**	90.696	98.643	37.098	88.354
I0.1-0.3-0.4	0.020	0.057	0.077	90.071	98.741	39.046	88.259
I0.1-0.3-0.6	0.021	0.057	0.078	89.014	98.571	36.518	88.482
G0.1	0.021	0.057	0.078	89.549	98.702	37.296	86.846
S0.1	0.019	0.057	**0.076**	**91.018**	98.759	46.266	89.652
S0.2	0.020	0.059	0.080	88.871	98.634	41.834	85.563
S0.3	0.019	0.064	0.083	87.523	98.616	42.100	89.086
Gauss0.1	**0.016**	0.059	**0.076**	90.259	**98.795**	**57.371**	**91.607**

Fig. 7. Qualitative results on our test set. A 3D coordinate axis (X, Y, Z axes) system is overlayed on the image based on the predicted pose. We provide more qualitative results in our supplementary material including results on real-data harvested from YouTube.

converge, in contrast with the higher λ_{exo} experiments for the IoU loss. These results show that the smoother silhouette consistency objective, independently of its implementation (i.e. Guassian or box-filter), behaves better than IoU and can improve the performance of direct pose estimation models when mask annotations are available. They also highlight the sensitivity in tuning the relative importance of the direct and silhouette based losses. Another important trait of the smoother objective is a smoother loss landscape, which is further analysed in the supplementary material, with qualitative results for in-the-wild data indicating a more robustly trained model as well.

6   Conclusions

In this work, we have filled in a data gap for the under-explored UAV-Assistant application context. We introduced a dataset that includes temporally and spatially aligned data, captured by the egocentric view of a cooperative drone, as well as the exocentric view of the user that observes the scene and the drone operating at her/his immediate surroundings. We carefully design a methodology to generate high-quality, multimodal data and then focus on an important aspect of such a cooperative framework, the spatial alignment between the user and the drone. Using data generated via synthesis, albeit realistic, we exploit the availability of drone silhouettes in the exocentric view and explore recent research in differentiable rendering to further improve pose estimation. We also introduce a smoother loss for learning rendered silhouettes and demonstrate its efficacy compared to the typical IoU loss. A parallel can be drawn between losses defined using distance fields with our smooth loss which, nonetheless, is quicker to evaluate and straightforward to implement compared to a symmetric distance field loss evaluation. Still, our works shows that smoother objectives are beneficial for projection based supervision, which warrants further exploration.

References

1. The 1st anti-uav challenge. In: the IEEE Conference on Computer Vision and Pattern Recognition Workshops (2020), https://anti-uav.github.io/, accessed on 03-03-2020
2. Antonini, A., Guerra, W., Murali, V., Sayre-McCord, T., Karaman, S.: The black-bird dataset: A large-scale dataset for uav perception in aggressive flight. arXiv preprint arXiv:1810.01987 (2018)
3. Belmonte, L.M., Morales, R., Fernández-Caballero, A.: Computer vision in autonomous unmanned aerial vehicles—a systematic mapping study. Applied Sciences **9**(15), 3196 (2019)
4. Bondi, E., Dey, D., Kapoor, A., Piavis, J., Shah, S., Fang, F., Dilkina, B., Hannaford, R., Iyer, A., Joppa, L., et al.: Airsim-w: A simulation environment for wildlife conservation with uavs. In: Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies. pp. 1–12 (2018)
5. Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., Rother, C.: Learning 6d object pose estimation using 3d object coordinates. In: European conference on computer vision. pp. 536–551. Springer (2014)
6. Brachmann, E., Michel, F., Krull, A., Ying Yang, M., Gumhold, S., et al.: Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3364–3372 (2016)
7. Chang, A., Dai, A., Funkhouser, T.A., Halber, M., Niebner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. In: 7th IEEE International Conference on 3D Vision, 3DV 2017. pp. 667–676. Institute of Electrical and Electronics Engineers Inc. (2018)
8. Chen, W., Gao, J., Ling, H., Smith, E., Lehtinen, J., Jacobson, A., Fidler, S.: Learning to predict 3d objects with an interpolation-based differentiable renderer. In: Advances In Neural Information Processing Systems (2019)
9. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289 (2015)
10. Coluccia, A., Fascista, A., Schumann, A., Sommer, L., Ghenescu, M., Piatrik, T., De Cubber, G., Nalamati, M., Kapoor, A., Saqib, M., et al.: Drone-vs-bird detection challenge at ieee avss2019. In: 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–7. IEEE (2019)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
12. Doumanoglou, A., Kouskouridas, R., Malassiotis, S., Kim, T.K.: Recovering 6d object pose and predicting next-best-view in the crowd. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3583–3592 (2016)
13. Erat, O., Isop, W.A., Kalkofen, D., Schmalstieg, D.: Drone-augmented human vision: Exocentric control for drones exploring hidden areas. IEEE transactions on visualization and computer graphics **24**(4), 1437–1446 (2018)
14. Fan, R., Jiao, J., Pan, J., Huang, H., Shen, S., Liu, M.: Real-time dense stereo embedded in a uav for road inspection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2019)
15. Fonder, M., Van Droogenbroeck, M.: Mid-air: A multi-modal dataset for extremely low altitude drone flights. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)

16. Friedman, J., Jones, A.C.: Fully automatic id mattes with support for motion blur and transparency. In: ACM SIGGRAPH 2015 Posters, pp. 1–1 (2015)

17. Guerra, W., Tal, E., Murali, V., Ryou, G., Karaman, S.: Flightgoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality. arXiv preprint arXiv:1905.11377 (2019)

18. Gupta, S., Arbeláez, P., Girshick, R., Malik, J.: Inferring 3d object pose in rgb-d images. arXiv preprint arXiv:1502.04652 (2015)

19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)

20. He, Y., Sun, W., Huang, H., Liu, J., Fan, H., Sun, J.: Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. arXiv preprint arXiv:1911.04231 (2019)

21. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In: Asian conference on computer vision. pp. 548–562. Springer (2012)

22. Hsieh, M.R., Lin, Y.L., Hsu, W.H.: Drone-based object counting by spatially regularized regional proposal networks. In: The IEEE International Conference on Computer Vision (ICCV). IEEE (2017)

23. Insafutdinov, E., Dosovitskiy, A.: Unsupervised learning of shape and pose with differentiable point clouds. In: Advances in neural information processing systems. pp. 2802–2812 (2018)

24. J., K., Smith, E., Lafleche, J.F., Fuji Tsang, C., Rozantsev, A., Chen, W., Xiang, T., Lebaredian, R., Fidler, S.: Kaolin: A pytorch library for accelerating 3d deep learning research. arXiv:1911.05063 (2019)

25. Jin, R., Jiang, J., Qi, Y., Lin, D., Song, T.: Drone detection and pose estimation using relational graph networks. Sensors **19**(6), 1479 (2019)

26. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1521–1529 (2017)

27. Kellenberger, B., Marcos, D., Tuia, D.: When a few clicks make all the difference: Improving weakly-supervised wildlife detection in uav images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)

28. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

29. Kisantal, M., Sharma, S., Park, T.H., Izzo, D., Märtens, M., D'Amico, S.: Satellite pose estimation challenge: Dataset, competition design and results. arXiv preprint arXiv:1911.02050 (2019)

30. Kosub, S.: A note on the triangle inequality for the jaccard distance. Pattern Recognition Letters **120**, 36–38 (2019)

31. Kyrkou, C., Theocharides, T.: Deep-learning-based aerial image classification for emergency response applications using unmanned aerial vehicles. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2019)

32. Li, Z., Wang, G., Ji, X.: Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 7678–7687 (2019)

33. Liao, S., Gavves, E., Snoek, C.G.M.: Spherical regression: Learning viewpoints, surface normals and 3d rotations on n-spheres. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
34. Loper, M.M., Black, M.J.: Opendr: An approximate differentiable renderer. In: European Conference on Computer Vision. pp. 154–169. Springer (2014)
35. Maggiori, E., Tarabalka, Y., Charpiat, G., Alliez, P.: Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In: IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE (2017)
36. Mahendran, S., Ali, H., Vidal, R.: 3d pose regression using convolutional neural networks. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 2174–2182 (2017)
37. Massa, F., Aubry, M., Marlet, R.: Convolutional neural networks for joint object detection and pose estimation: A comparative study. arXiv preprint arXiv:1412.7190 (2014)
38. Mueller, M., Smith, N., Ghanem, B.: A benchmark and simulator for uav tracking. In: Proc. of the European Conference on Computer Vision (ECCV) (2016)
39. Müller, M., Casser, V., Lahoud, J., Smith, N., Ghanem, B.: Sim4cv: A photo-realistic simulator for computer vision applications. International Journal of Computer Vision **126**(9), 902–919 (2018)
40. Oh, S., Hoogs, A., Perera, A., Cuntoor, N., Chen, C.C., Lee, J.T., Mukherjee, S., Aggarwal, J., Lee, H., Davis, L., et al.: A large-scale benchmark dataset for event recognition in surveillance video. In: CVPR 2011. pp. 3153–3160. IEEE (2011)
41. Palazzi, A., Bergamini, L., Calderara, S., Cucchiara, R.: End-to-end 6-dof object pose estimation through differentiable rasterization. In: The European Conference on Computer Vision (ECCV) Workshops (September 2018)
42. Park, K., Mousavian, A., Xiang, Y., Fox, D.: Latentfusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation. arXiv preprint arXiv:1912.00416 (2019)
43. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
44. Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H.: Pvnet: Pixel-wise voting network for 6dof pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4561–4570 (2019)
45. Pepik, B., Gehler, P., Stark, M., Schiele, B.: 3d 2 pm–3d deformable part models. In: European Conference on Computer Vision. pp. 356–370. Springer (2012)
46. Periyasamy, A.S., Schwarz, M., Behnke, S.: Refining 6d object pose predictions using abstract render-and-compare. arXiv preprint arXiv:1910.03412 (2019)
47. Proenca, P.F., Gao, Y.: Deep learning for spacecraft pose estimation from photo-realistic rendering. arXiv preprint arXiv:1907.04298 (2019)
48. Rad, M., Lepetit, V.: Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3828–3836 (2017)
49. Rambach, J., Deng, C., Pagani, A., Stricker, D.: Learning 6dof object poses from synthetic single channel images. In: 2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). pp. 164–169. IEEE (2018)
50. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 658–666 (2019)

51. Robicquet, A., Alahi, A., Sadeghian, A., Anenberg, B., Doherty, J., Wu, E., Savarese, S.: Forecasting social navigation in crowded complex scenes. arXiv preprint arXiv:1601.00998 (2016)
52. Rozantsev, A., Lepetit, V., Fua, P.: Detecting flying objects using a single moving camera. IEEE transactions on pattern analysis and machine intelligence **39**(5), 879–892 (2016)
53. Schwarz, M., Schulz, H., Behnke, S.: Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features. In: 2015 IEEE international conference on robotics and automation (ICRA). pp. 1329–1335. IEEE (2015)
54. Shah, S., Dey, D., Lovett, C., Kapoor, A.: Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: Field and service robotics. pp. 621–635. Springer (2018)
55. Shi, X., Yang, C., Xie, W., Liang, C., Shi, Z., Chen, J.: Anti-drone system with multiple surveillance technologies: Architecture, implementation, and challenges. IEEE Communications Magazine **56**(4), 68–74 (2018)
56. Su, H., Qi, C.R., Li, Y., Guibas, L.J.: Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2686–2694 (2015)
57. Tejani, A., Kouskouridas, R., Doumanoglou, A., Tang, D., Kim, T.K.: Latent-class hough forests for 6 dof object pose estimation. IEEE transactions on pattern analysis and machine intelligence **40**(1), 119–132 (2017)
58. Tekin, B., Sinha, S.N., Fua, P.: Real-time seamless single shot 6d object pose prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 292–301 (2018)
59. Tulsiani, S., Malik, J.: Viewpoints and keypoints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1510–1519 (2015)
60. Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., Savarese, S.: Densefusion: 6d object pose estimation by iterative dense fusion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3343–3352 (2019)
61. Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J.: Normalized object coordinate space for category-level 6d object pose and size estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2642–2651 (2019)
62. Wang, M., Deng, W.: Deep visual domain adaptation: A survey. Neurocomputing **312**, 135–153 (2018)
63. Wu, Y., Marks, T., Cherian, A., Chen, S., Feng, C., Wang, G., Sullivan, A.: Unsupervised joint 3d object model learning and 6d pose estimation for depth-based instance segmentation. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 0–0 (2019)
64. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes (2018)
65. Xie, K., Yang, H., Huang, S., Lischinski, D., Christie, M., Xu, K., Gong, M., Cohen-Or, D., Huang, H.: Creating and chaining camera moves for quadrotor videography. ACM Transactions on Graphics (TOG) **37**(4), 1–13 (2018)
66. Yuan, L., Reardon, C., Warnell, G., Loianno, G.: Human gaze-driven spatial tasking of an autonomous mav. IEEE Robotics and Automation Letters **4**(2), 1343–1350 (April 2019). https://doi.org/10.1109/LRA.2019.2895419

67. Zakharov, S., Shugurov, I., Ilic, S.: Dpod: 6d pose object detector and refiner. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1941–1950 (2019)
68. Zhang, X., Jia, N., Ivrissimtzis, I.: A study of the effect of the illumination model on the generation of synthetic training datasets. arXiv preprint arXiv:2006.08819 (2020)
69. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
70. Zhu, P., Wen, L., Bian, X., Ling, H., Hu, Q.: Vision meets drones: A challenge. arXiv preprint arXiv:1804.07437 (2018)