Wide-Depth-Range 6D Object Pose Estimation in Space

Yinlin Hu [1], Sébastien Speierer [2], Wenzel Jakob [2], Pascal Fua [1], Mathieu Salzmann [1,3]
[1] EPFL Computer Vision Lab, [2] EPFL Realistic Graphics Lab, [3] ClearSpace SA
{firstname.lastname}@epfl.ch

Abstract

6D pose estimation in space poses unique challenges that are not commonly encountered in the terrestrial setting. One of the most striking differences is the lack of atmospheric scattering, allowing objects to be visible from a great distance while complicating illumination conditions. Currently available benchmark datasets do not place a sufficient emphasis on this aspect and mostly depict the target in close proximity.

Prior work tackling pose estimation under large scale variations relies on a two-stage approach to first estimate scale, followed by pose estimation on a resized image patch. We instead propose a single-stage hierarchical end-to-end trainable network that is more robust to scale variations. We demonstrate that it outperforms existing approaches not only on images synthesized to resemble images taken in space but also on standard benchmarks.

(a) (b) (c)

Figure 1: **Docking and space cleaning. (a, b)** Two different views of the Agena target vehicle during the first space docking. The appearance of Agena is strongly affected by the large scale and viewpoint changes, suggesting that different image features should be used for 6D pose estimation. In 1966, this docking procedure was controlled manually. **(c)** In 2025, the ClearSpace One chaser satellite will be launched to retrieve and de-orbit a non-operational satellite, so as to showcase the feasibility of removing space debris. In this case, the capture will be fully automated. The synthetic image shown here highlights the challenges the algorithm will have to handle, such as reflections, over-exposure of some parts of the images, and lack of details in others.

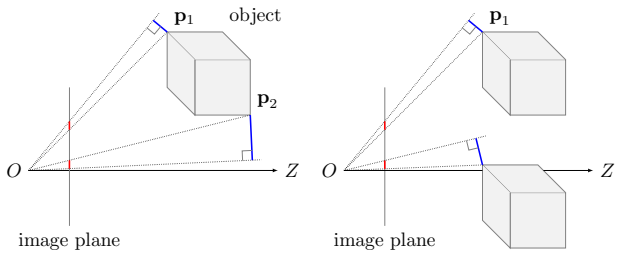
1. Introduction

Reliable 6D pose estimation is key to automating many spatial maneuvers, such as docking or capturing inert objects as shown in Fig. 1. An important consequence of such maneuvers is that they dramatically change the scale and aspect of the observed target. Although 6D pose estimation is an active area of research in computer vision and robotics, this important aspect has not received significant attention thus far—for example, most benchmark datasets [8, 20, 42, 9] feature objects whose depth varies within a limited range. The lack of atmospheric scattering enabling observation from great distances also leads to other challenges: harsh contrast, under- and over-exposed areas, and significant specular reflections from reflective materials used in space engineering (aluminium and carbon fiber panels, etc.).

To address such challenges, the European Space Agency (ESA) and Stanford University recently organized a satellite pose estimation challenge based on the *Spacecraft Pose Estimation Dataset* (SPEED) [19]. The best-performing methods in this competition use a two-step approach to handle large depth variation: a detector finds an axis-aligned box bounding the target, which is resampled to a uniform size and finally processed by a 6D pose estimator.

This approach is suboptimal in several ways. First, detection and pose estimation are treated as separate processes, which precludes joint training. Second, it provides supervisory signals only to the final layer of the encoder-decoder architecture being used instead of to all levels of the decoding pyramid, which would increase robustness. Third, many similar feature extraction computations are performed by both processes, which results in an unnecessary duplication of effort. Finally, these methods rely on the dominant approach to deep learning based 6D object pose estimation [33, 11, 2] consisting of training a network to minimize the 2D reprojection error of predefined 3D keypoints, which cannot cope with large depth range variations: As shown in Fig. 2, reprojection error is strongly affected by the distance of individual keypoints to the camera, and not explicitly taking this into account degrades performance.

To address these shortcomings, we introduce a single hierarchical end-to-end trainable network depicted by Fig. 3 that yields robust and scale-insensitive 6D poses.



(a) Sensitivity to different keypoints (b) Sensitivity to target positions

Figure 2: **Problem with minimizing the 2D reprojection error. (a)** The red lines denote the 2D reprojection errors for points $\mathbf{p}_1$ and $\mathbf{p}_2$. Because one is closer to the camera than the other, these 2D errors are of about the same magnitude even though the corresponding 3D errors, shown in blue, are very different. **(b)** For the same object at different locations, the same 2D error can generate different 3D errors. This makes pose accuracy dependent on the relative position of the target to the camera.

To use information across scales, it progressively downscales the learned features, derives 3D-to-2D correspondences for each level of the resulting pyramid, and finally uses a RANSAC-based PnP strategy to infer a single reliable pose from these sets of correspondences. This is a departure from most networks that estimate pose only from the final layer. To address the issue in Fig. 2, we minimize a training loss based on 3D positions instead of 2D projections, making the method invariant to the target distance. We use a Feature Pyramid Network (FPN) [24] as our backbone but, unlike in most approaches relying on such networks, we assign each training instance to multiple pyramid levels to promote the joint use of multi-scale information.

In short, our contribution is a new 6D pose estimation architecture that reliably handles large scale changes under challenging conditions. We will show that it outperforms all state-of-the-art methods on the established SPEED dataset while also being much faster. Furthermore, we introduce a larger-scale satellite pose estimation dataset featuring more realistic and more complex images than SPEED, and we show that our method delivers the same benefits in this more challenging scenario. Finally, we demonstrate that our method outperforms the state of the art even on images with smaller depth variations, such as those of the challenging Occluded LINEMOD dataset. Our code and new dataset will be publicly released.

2. Related Work

The most commonly-used sensors for 6D pose estimation in space remain cameras, may they be RGB, monochromatic, or, although more rarely, infrared. We therefore focus on image-based 6D pose estimation in both our work and the discussion below.

The standard framework to perform 6D pose estimation

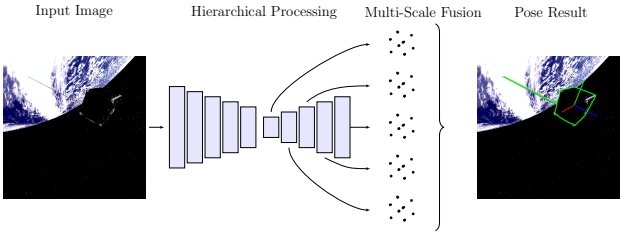
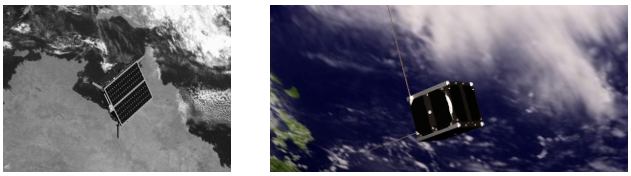


Figure 3: **Our single-stage approach.** We use an encoder-decoder architecture to progressively downsample the image and then to re-expand it. At each level of the decoder, we establish 3D-to-2D correspondences. Finally, we use a RANSAC-based PnP strategy [21] to infer a single reliable pose from these sets of correspondences.

consists of first establishing 3D-to-2D correspondences, and then compute the pose using a PnP solver [27, 41, 30]. While many hancrafted methods have been designed to extract the required correspondences [26, 39, 40], they tend to produce low-quality output under challenging conditions (objects lacking spatial variation, strong highlights, etc.). As such, most modern 6D object pose estimation methods establish such correspondences using a neural network. This network is usually trained to predict the image location of the 3D object bounding box corners, either in a single global fashion [18, 33, 37, 42], or by aggregating multiple local predictions to improve robustness to occlusions [29, 16, 11, 31, 43, 23]. Whether global or local, these methods were designed to be effective on standard computer vision benchmarks, which feature minimal scale changes. As we will show in our experiments, they therefore perform poorly when the depth range at which the object is depicted varies dramatically across different images.

The few works that have attempted to handle the scale issue rely on an object detection network as a preprocessing component [22, 23, 2]. While the zoom sampling strategy introduced in [23] aims to account for the object detection noise when training the pose network, it still does not reflect the true distribution of the patches output by the detection network, and the resulting framework does not unify the detection and pose estimation stages. While this could in principle be achieved via a Spatial Transformer Network [15], such a change would significantly complicate the architecture, introducing redundant operations across the detection and pose estimation modules and eventually precluding real-time inference. Our main contribution entails using the inherent hierarchical structure of a single network with shared weights across the levels to handle the scale problem. We demonstrate this to be both robust and efficient.

Hierarchical processing, such as image pyramids [1, 13, 17], is a classical idea for multi-scale image understanding [14, 12]. Recently, this idea has been translated to the deep learning realm via Feature Pyramid Networks (FPNs) [24], which are now a standard component of many object detection frameworks [25, 38, 45]. Here, we lever-



(a) The SPEED dataset (b) The proposed SwissCube dataset

Figure 4: **Comparison of datasets. (a)** The SPEED dataset [19] was generated with a non-physics-based renderer and only poorly reflects the complexity of illumination in space. **(b)** We introduce a SwissCube dataset that was created via physics-based rendering.

age this idea for 6D object pose estimation. However, unlike most object detection methods that explicitly associate each pyramid level to a single, predefined scale, we introduce a dynamic sampling strategy where each training instance leverages all pyramid levels, albeit with different weights. This allows us to fuse the predictions from the different levels at inference, leading to more robust 6D pose estimates.

We focus our experiments on 6D pose estimation of space-borne objects, because robustness to scale is highly important in that context, particularly when approaching non-cooperative targets (e.g. space trash) that require motion synchronization. The space engineering community has its own literature on the topic of 6D pose estimation. While it has evolved in a manner that resembles progress in computer vision, it has mostly focused on handcrafted methods [44, 5, 32, 35], with only a few works proposing deep learning based approaches [2]. The main reason for this is the lack of large amounts of annotated data for space-borne objects. Recently, this was addressed by the SPEED dataset [19] released by ESA and Stanford University as part of a satellite pose estimation challenge. This dataset, however, has several limitations. First, it does not provide the 3D model of the satellite, and while it can be reconstructed from the images, the final pose estimate will depend not only on the pose estimation algorithm but also on the quality of this reconstruction. Second, the SPEED images were synthesized by a non-physics-based rendering technique, only poorly reflecting the complexity of illumination in space, as illustrated in Fig. 4. Finally, the depth distribution of the SPEED dataset is not uniform, with only few images depicting the satellite at a large distance from the camera. However, accurate pose for farther objects can be critical for space rendezvous; they give the docker or chaser enough time to adjust its own motion and prepare for the actual operations. We propose a novel satellite pose estimation dataset that addresses this bias, and constitutes the second contribution of this article. The images in this dataset were created using a physically-based spectral light transport simulation involving an accurate reference 3D model of a cube satellite that accounts for the effects of the Sun, Earth, stars, etc.

3. Approach

Our goal is to estimate the 3D rotation and 3D translation of a known rigid object depicted in an RGB image. To this end, we design a deep network that regresses the 2D projections of predefined 3D points. However, rather than regressing the 2D projections at a single, fixed scale, which lacks robustness to large depth variations, we use a Feature Pyramid Network (FPN) [24], perform the regression at multiple scales, and fuse the resulting multiple estimates in a robust pose prediction.

In the following sections, we first present the FPN architecture our network builds on and then introduce a sampling-based training strategy to leverage every pyramid level for each training instance. Finally, we discuss our fusion approach to obtaining a single pose estimate during inference.

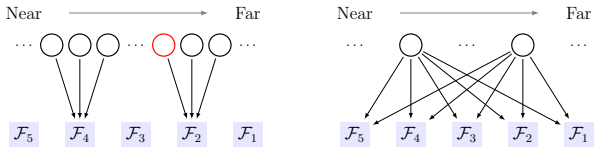
3.1. Pyramid Network Architecture

Most 6D pose estimation deep networks rely on an encoder-decoder architecture. Therefore, to handle large scale variations for 6D object pose estimation, instead of relying on an additional object detection network, we use the inherent hierarchical architecture of the encoder network, which extracts features at different scales. Specifically, we use Darknet-53 [34] as backbone in our framework and employ the same network architecture as in the FPN [24] designed for object detection, which consists of $k = 5$ levels of feature maps, $\{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4, \mathcal{F}_5\}$, each with an increasingly large receptive field.

Instead of computing a single pose estimate from the feature map $\mathcal{F}_5$ only, we regress the 2D locations of the object 3D keypoints from every level of this pyramid. To this end, we rely on the segmentation-driven approach of [11], and make the feature vector at every spatial location in each feature map to output the 2D projections of the 3D keypoints, represented as an offset from the center of the corresponding cell, and an objectness score for each object class. The feature vector at each cell therefore is a $C \times (2 \times 8 + 1)$ dimensional vector consisting of 8 2D offsets and an objectness indicator for C object classes. To encode a segmentation mask, all feature cells need to be involved in the objectness prediction, including those that contain no target objects. By contrast, as discussed below, only selected cells are involved in training the pose regressor.

3.2. Ensemble-Aware Sampling

Large-scale variations impose drastic difficulties on the network for accurate prediction for every scale. The standard approach to training an FPN follows a divide-and-conquer strategy, consisting of dividing the whole training set of instances into several non-overlapping groups according to the object size and then assigning different groups



(a) Standard strategy [24] (b) Proposed strategy

Figure 5: **Sampling strategies during training.** Let the circles denote all the training instances sorted in increasing order of depth from left to right. **(a)** The traditional sampling strategy assigns each instance to a single pyramid level according to its size during training. For example, the red instance is fed only to pyramid level $\mathcal{F}_2$, thus encouraging only this level to yield a reasonable prediction for this sample. **(b)** We propose to assign each instance to multiple pyramid levels, encouraging every pyramid level to produce a reasonable pose estimate for every instance.

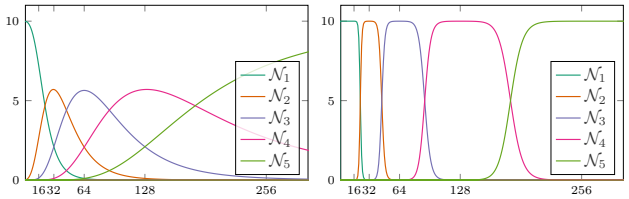


Figure 6: **Sample count $\mathcal{N}_k$ at each pyramid level as a function of the object size $\mathcal{S}$.** Typically, when $\lambda > 20$, $\mathcal{N}_k$ degenerates to the simple "hard" assignment strategy that FPN adopts. Note that for a given object size, multiple $\mathcal{N}_k$s are non-zero, which translates to soft assignments to the different pyramid levels.

to different pyramid levels during training, as illustrated in Fig. 5(a). This simple strategy may be sufficient for object detection where one can simply choose level producing the best prediction based on the objectness scores during testing. However, for 6D pose estimation, it prevents one from leveraging the predictions of the multiple levels jointly to improve robustness, because, for a given scale, most levels will yield highly noisy estimates as they weren't trained for objects at that scale.

To address this issue, we design a sampling strategy that allows every feature vector within the object segmentation mask at each level to participate in the prediction with a certain probability, as in Fig. 5(b). Let s_k, for $1 \leq k \leq 5$, be a reference object size for level k of the pyramid, chosen based on the object size distribution in the target dataset. For example, in our SwissCube dataset, we take s_k to be $16, 32, 64, 128$, and 256, respectively. Then, for an object of size $\mathcal{S}$ taken to be the largest of the width and height of its 2D bounding box, we uniformly randomly sample

$$\mathcal{N}_k = \alpha \frac{e^{-\lambda \Delta_k^2}}{\sum_{j=1}^{5} e^{-\lambda \Delta_j^2}} \tag{1}$$

feature vectors at level k among those within the object segmentation mask, with

$$\Delta_k = |\log_2 \frac{\mathcal{S}}{s_k}| \quad \text{and} \quad \alpha = 10\ . \tag{2}$$

The hyper-parameter α specifies the maximum number of active feature vectors on any level, and $\lambda \geq 0$ controls the distribution of the number of active cells across levels. When $\lambda = 0$, all $\mathcal{N}_k$s are equal, thus using the same number of feature cells at each pyramid level, independently of the object size. By contrast, when λ is large, that is, $\lambda > 20$, the sampling strategy degenerates to the "hard assignment" commonly-used by FPNs. In Fig. 6, we show how each $\mathcal{N}_k$ varies as a function of $\mathcal{S}$ for different λ values. Note that, for a given object size, multiple pyramid levels will be involved in training, thus making them robust to scale variations.

3.3. Loss Function in 3D Space

As mentioned before, every feature vector selected by our sampling procedure is then used to regress the 2D projections of the 8 corners of the 3D object bounding box. When regressing 2D locations, most existing methods [33, 11] seek to directly minimize the error in the image plane, that is, the loss function $\sum_{i=1}^{n} |\mathbf{u}_i - \hat{\mathbf{u}}_i|$, where $\mathbf{u}_i$ is the ground-truth 2D projection and $\hat{\mathbf{u}}_i$ the predicted one. However, as illustrated by Fig. 2, this loss function is suboptimal, particularly in the presence of large depth variations, because it puts more emphasis on some keypoints than on others and also depends on the object's relative position.

To overcome this, we introduce a loss function in 3D space, which is invariant to the depth of 3D keypoints. Under a perspective camera model, the projection of a 3D object keypoint $\mathbf{p}_i$ in the image is given by

$$\lambda_i \begin{bmatrix} \mathbf{u}_i \\ 1 \end{bmatrix} = \mathbf{K}(\mathbf{R}\mathbf{p}_i + \mathbf{t}), \tag{3}$$

where $\mathbf{u}_i$ is the 2D image location, λ_i is a scale factor, $\mathbf{K}$ is the 3×3 matrix of camera intrinsic parameters, and $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and translation vector representing the 6D object pose. Then, let

$$\hat{\mathbf{v}}_i = \mathbf{K}^{-1}[\hat{u}_i, \hat{v}_i, 1]^\top \tag{4}$$

$$\mathbf{p}_i^c = \mathbf{R}\mathbf{p}_i + \mathbf{t} \tag{5}$$

be the 3D camera ray passing through the predicted 2D location $\hat{\mathbf{u}}_i = [\hat{u}_i, \hat{v}_i]$ and the corresponding 3D keypoint $\mathbf{p}_i$ expressed in the camera coordinate system, respectively, where $\mathbf{R}$ and $\mathbf{t}$ are the ground-truth rotation matrix and translation vector. We can then map the re-projection error into 3D space by computing

$$\begin{aligned} \mathbf{e}_i &= \mathbf{p}_i^c - \hat{\mathbf{V}}_i \mathbf{p}_i^c \\ &= (\mathbf{I} - \hat{\mathbf{V}}_i)\mathbf{p}_i^c\ , \end{aligned} \tag{6}$$

where

$$\hat{\mathbf{V}}_i = \frac{\hat{\mathbf{v}}_i \hat{\mathbf{v}}_i^\top}{\hat{\mathbf{v}}_i^\top \hat{\mathbf{v}}_i} \tag{7}$$

is a matrix projecting a 3D point orthogonally to the camera ray $\hat{\mathbf{v}}_i$ [27], as illustrated in Fig. 2. Finally, we take our pose regression loss to be

$$\mathcal{L}_{reg} = \sum_{i=1}^{n} sl_1(\mathbf{e}_i). \tag{8}$$

where $sl_1(\cdot)$ is the smoothed L1 norm [6]. As shown in Fig. 10, this 3D error is consistent across all 3D keypoints and less influenced by the depth and relative position of the observed object. Furthermore, it can be computed by simple algebraic operations and can thus easily be incorporated in an end-to-end learning formalism.

Ultimately, we combine this loss function with that supervising the predicted objectness score, which yields the overall training loss

$$\mathcal{L} = \sum_{k=1}^{5} \{\mathcal{L}_{obj}(k) + \mathcal{L}_{reg}(k)\}, \tag{9}$$

where $\mathcal{L}_{obj}(k)$ and $\mathcal{L}_{reg}(k)$ are the objectness loss and pose regression loss at level k, respectively. In this work, we take the loss $\mathcal{L}_{obj}$ to be the focal loss [25].

3.4. Inference via Multi-Scale Fusion

Thanks to our ensemble-aware sampling strategy, our trained network can produce valid pose estimates at every pyramid level for any test image, independently of its scale. These estimates can be selected by thresholding the objectness score predicted for each feature vector at each level, and in practice we use a threshold $\tau = 0.3$. In principle, these estimates could then be fused directly by a RANSAC+PnP strategy [21] or using the learning-based method of [10]. For simplicity, we use the RANSAC+PnP approach, but in conjunction with our ensemble-aware sampling scheme.

To apply this scheme at test time, we first need to estimate the object size. To this end, we choose the feature vector leading to the highest objectness score, and compute the size $\mathcal{S}$ from the corresponding predictions of the 8 bounding box corner projections. Given this size, we then select, for each pyramid level k, the $\mathcal{N}_k$ feature cells that give the highest objectness score. This lets us construct a set of 3D-to-2D correspondences $\{\mathbf{p}_i \leftrightarrow \mathbf{u}_{ijk}\}$ for every 3D keypoint $\mathbf{p}_i$, where $\mathbf{u}_{ijk}$ is the 2D location predicted for $\mathbf{p}_i$ by cell $\mathcal{C}_j$ on feature map $\mathcal{F}_k$, with $1 \leq i \leq 8, 1 \leq j \leq \mathcal{N}_k$ and $1 \leq k \leq 5$. Finally, we use a RANSAC based PnP algorithm to obtain a robust 6D pose estimate from these correspondences. We will show in our experiments that this outperforms the prediction obtained from any individual pyramid level.

4. Experiments

In this section, we first evaluate our framework on the SPEED dataset, and then introduce the SwissCube dataset, which contains accurate 3D mesh and physically-modeled astronomical objects, and perform thorough ablation studies on it. We further show results on real images of the same satellite. Finally, to demonstrate the generality of our approach we evaluate it on the standard Occluded-LINEMOD dataset depicting small depth variations.

We train our model starting from a backbone pre-trained on ImageNet [4], and, for any 6D pose dataset, feed it 3M unique training samples obtained via standard online data augmentation strategies, such as random shift, scale, and rotation. To evaluate the accuracy, we will report the individual performance under different depth ranges, using the standard ADI-0.1d [11, 10] accuracy metrics, which encodes the percentage of samples whose 3D reconstruction error is below 10% of the object diameter. On the SPEED dataset, however, we use a different metric, as we do not have access to the 3D SPEED model, making the computation of ADI impossible. Instead, we use the metric from the competition, that is, $\mathbf{e_q} + \mathbf{e_t}$, where $\mathbf{e_q}$ is the angular error between the ground-truth quaternion and the predicted one, and $\mathbf{e_t}$ is the normalized translation error. Furthermore, because the depth distribution of SPEED is not uniform, with only few images depicting the satellite at a large distance from the camera, we only report the average error on the whole test set, as in the competition. The source code and dataset are publicly available at https://github.com/cvlab-epfl/wide-depth-range-pose.

4.1. Evaluation on the SPEED Dataset

Although the SPEED dataset has several drawbacks, discussed in Section 2, it remains a valuable benchmark, and we thus begin by evaluating our method on it. As the test annotations are not publicly available, and the competition is not ongoing, we divide the training set into two parts, 10K images for training and the remaining 2K ones for testing. We evaluate the two top-performing methods from the competition, [2] (DLR) and [11] (SegDriven-Z), on these new splits using the publicly-available code, and find their errors to be of similar magnitude to the ones reported online during the challenge. Note that our method, as DLR and SegDriven-Z, uses the 3D model to define the keypoints whose image location we predict. We therefore exploit a method of [7] to first reconstruct the satellite from the dataset.

Table 1 compares our results to those of the two top-performing methods on this dataset. Note that DLR combines the results of 6 pose estimation networks, followed by an additional pose refinement strategy to improve accuracy. We therefore also report the results of our method with and without this pose refinement strategy. Note, however, that

		Accuracy		Model Size	FPS
		Raw	Refinement		
SegDriven-Z [11]		0.022	-	89.2 M	3.1
DLR [2]		0.017	0.012	176.2 M	0.7
Ours	640×	0.018	0.013	**51.5 M**	**35**
	960×	**0.016**	**0.010**	**51.5 M**	18

Table 1: **Comparison with the state of the art on SPEED.** Our method outperforms the two top-performing methods in the challenge and is much faster and lighter.

we still use a single pose estimation network. Furthermore, for our method, we report the results of two separate networks trained at different input resolutions. At the resolution of 960×, we outperform the two state-of-the-art methods, while our architecture is much smaller and much faster. To further speed up our approach, we train a network at a third (640×) of the raw image resolution. This network remains on par with DLR but runs 20+ times faster.

4.2. Evaluation on the SwissCube Dataset

To facilitate the evaluation of 6D object pose estimation methods in the wide-depth-range scenario, we introduce a novel SwissCube dataset. The renderings in this dataset account for the precise 3D shape of the satellite and include realistic models of the star backdrop, Sun, Earth, and target satellite, including the effects of global illumination, mainly glossy reflection of the Sun and Earth from the satellite's surface. To create the 3D model of the SwissCube, we modeled every mechanical part from raw CAD files, including solar panels, antennas, and screws, and we carefully assigned material parameters to each part.

The renderings feature a space environment based on the relative placement and sizes of the Earth and Sun. Correct modeling of the Earth is most important, as it is often directly observed in the images and significantly affects the appearance of the satellite via inter-reflection. We extract a high-resolution spectral texture of the Earth's surface and atmosphere from published data products acquired by the NASA Visible Infrared Imaging Radiometer Suite (VIIRS) instrument. These images account for typical cloud coverage and provide accurate spectral color information on 6 wavelength bands. Illumination from the Sun is also modeled spectrally using the extraterrestrial solar irradiance spectrum. The spectral simulation performed using the open source Mitsuba 2 renderer [28] finally produces an RGB output that can be ingested by standard computer vision tools.

The renderings also include a backdrop of galaxies, nebulae, and star clusters based on the HYG database star catalog [3] containing around 120K astronomical objects along with information about position and brightness. The irradiance due to astronomical objects is orders of magnitude below that of the Sun. To increase the diversity of the dataset,

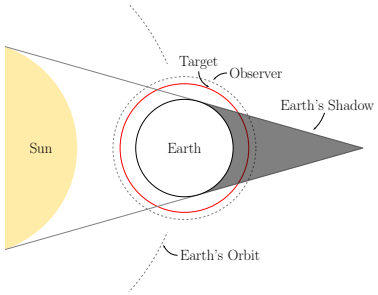


Figure 7: **Settings for physical rendering of SwissCube.** We physically model the Sun, the Earth, and the complex illumination conditions that can occur in space.

and to ensure that the network ultimately learns to ignore such details, we boost the brightness of astronomical objects in renderings to make them more apparent.

Following these steps, we place the SwissCube into its actual orbit located approximately 700 km above the Earth's surface along with a virtual observer positioned in a slightly elevated orbit. We render sequences with different relative velocities, distances and angles. To this end, we use a wide field-of-view (100°) camera whose distance to the target ranges uniformly between $1d$ to $10d$, where d indicates the diameter of the SwissCube without taking the antennas into accounts. The high-level setup is illustrated in Fig. 7. Note that the renderings are essentially black when the SwissCube passes into the earth's shadow, and we detect and remove such configurations.

We generate 500 scenes each consisting of a 100-frame sequence, for a total of 50K images. We take 40K images from 400 scenes for training and the 10K image from the remaining 100 scenes for testing. We render the images at a 1024×1024 resolution, a few of which are shown in Fig. 8. During network processing, we resize the input to 512×512. We report the ADI-0.1d accuracy at three depth ranges, which we refer to as *near*, *medium*, and *far*, corresponding to the depth ranges [1d-4d], [4d-7d], and [7d-10d], respectively.

4.2.1 Effect of our Ensemble-Aware Sampling

We first evaluate the effectiveness of our ensemble-aware sampling strategy, further comparing our approach with the single-scale baseline SegDriven [11], which uses the same backbone as us. Note that the original SegDriven method did not rely on a detector to zoom in on the object, but was extended with a YOLOv3 [34] one in the SPEED competition, resulting in the SegDriven-Z approach evaluated above. For our comparison on the SwissCube dataset to be fair, we therefore also report the results of SegDriven-Z. Moreover, we also evaluate the top performer on the SPEED dataset, DLR [2], on our dataset.

Fig. 9 demonstrates the effectiveness of our sampling

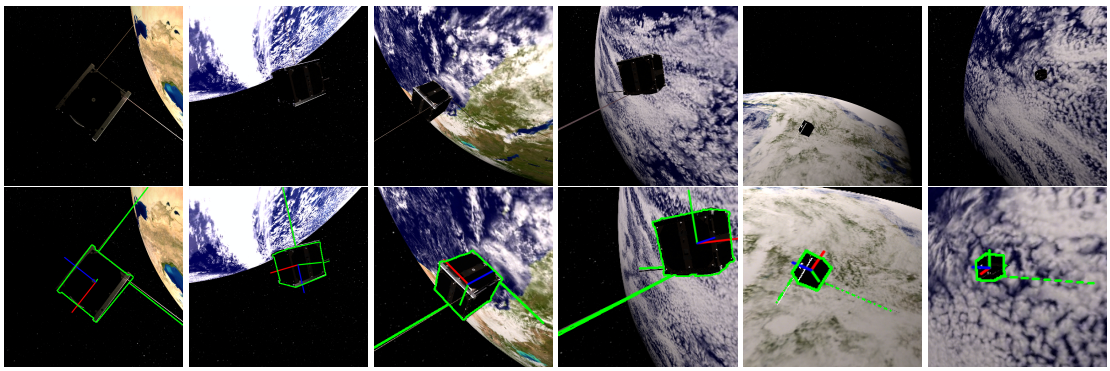


Figure 8: **Qualitative results on the SwissCube dataset.** Our method yields accurate pose estimates at all scales.

	Near	Medium	Far	All
SegDriven [11]	41.1	22.9	7.1	21.8
SegDriven-Z [11]	52.6	45.4	29.4	43.2
DLR [2]	63.8	47.8	28.9	46.8
Ours	**65.2**	**48.7**	**31.9**	**47.9**

Table 2: **Our method outperforms all baselines on SwissCube.**

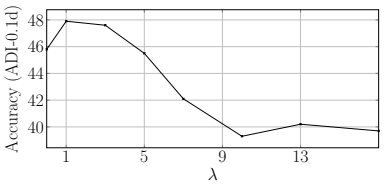


Figure 9: **Effect of ensemble-aware sampling.** In general, the more cross-level samples are involved in training, that is, the smaller λ is, the better the results.

strategy. Our results with different λ values, which controls the ensemble-aware sampling, show that large values, such as $\lambda > 10$, yield lower accuracies. With such large values, our sampling strategy degenerates to the one commonly-used in FPN-based object detectors. This therefore evidences the importance of encouraging every pyramid level to produce valid estimates at more than a single object scale. Note also that $\lambda = 0$, which corresponds to distributing every training instance uniformly to all levels, does not yield the best results, suggesting that forcing every level to produce high-accuracy at all the scales is sub-optimal. In other words, each level should perform well in a reasonable scale range, but these ranges should overlap across the pyramid levels. This is achieved approximately with $\lambda = 1$, which we will use in the following experiments.

Table 2 summarizes the comparison results with other baselines. Because it does not explicitly handle scale, SegDriven performs poorly on far objects. This is improved by the detector used in SegDiven-Z. However, the performance of this two-stage approach remains much worse than that of our framework. Our method outperforms DLR as well, even though our method is 20+ times faster than DLR. Fig. 8 depicts a few rendered images and corresponding poses estimated with our approach.

4.2.2 Effect of our Multi-Scale Fusion

To better understand the role of each pyramid level during multi-scale fusion, we study the accuracy obtained using the predictions of each individual pyramid level. Intuitively, we expect the levels with a larger receptive field (feature maps with low spatial resolution) to perform well for close objects, and those with a small receptive field (feature maps with high spatial resolution) to produce better results faraway ones. While the results in Table 3 confirm this intuition for Levels L1, L2 and L3, we observe that the performance degrades at L4 and L5. We believe this to be due to the very low spatial resolution of the corresponding feature maps, 8×8, and 4×4, respectively, making it difficult for these levels to output precise poses. Nevertheless, the accuracy after multi-scale fusion outperforms every individual level, and we leave the study of a different number of pyramid levels to future work.

	Near	Medium	Far	All
L1	0	25.2	31.8	19.5
L2	36.5	48.4	27.7	38.2
L3	62.3	47.4	19.9	42.6
L4	59.2	20.2	1.7	26.3
L5	25.5	0.9	0	8.3
Fusion	**65.2**	**48.7**	**31.9**	**47.9**

Table 3: **Effect of the multi-scale fusion.** Each pyramid level favors a specific depth range, which our multi-scale fusion strategy leverages to outperform every individual level.

4.2.3 Effect of the 3D Loss

In Table 4, we compare the results obtained by training our approach with either the commonly-used 2D reprojection loss or our loss function in 3D space. Note that our 3D loss outperforms the 2D one in all depth ranges, and the farther the object, the larger the gap between the results of the two

	Near	Medium	Far	All
2D loss	64.6	42.0	24.0	43.1
3D loss	**65.2**	**48.7**	**31.9**	**47.9**
Delta	+0.6	+6.7	+7.9	+4.8

Table 4: **Effect of the 3D loss.** The proposed 3D loss outperforms the 2D one in every depth ranges. The farther the object, the more obvious the advantage of the 3D loss.

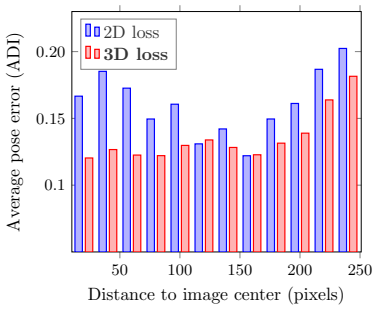


Figure 10: **Pose error as a function of the object position.** The performance of the 2D loss clearly degrades for objects near the image center, whereas that of our 3D loss doesn't. See Fig. 2(b) for the underlying geometry. Note that as the object moves closer to the image boundary, it becomes truncated, which degrades the performance of both losses.

loss functions. In Fig. 10, we plot the average accuracy as a function of the object image location. The performance of the 2D loss degrades significantly when the object is located near the image center, whereas the accuracy of our 3D loss remains stable for most object positions. Note that, The reason both of them become worse in the right part of the figure is due to the object truncation by image borders.

4.3. Results on Real Images

In Fig. 11, we illustrate the performance of our approach on real images. Note that these real images were not captured in space but in a lab environment using a mock-up model of the target and an OptiTrack motion capture system to obtain ground-truth pose information for a few images. We then fine-tuned our model pre-trained on our synthetic SwissCube dataset using only 20 real images with pose annotations. Because this procedure only requires small amounts of annotated real data, it would be applicable in an actual mission, where images can be sent to the ground, annotated manually, and the updated network parameters uploaded back to space.

4.4. Evaluation on Occluded-LINEMOD

Finally, to demonstrate that our approach is general, and thus applies to datasets depicting small depth variations, we evaluate it on the standard Occluded-LINEMOD dataset [20]. Following [10], we use the raw images at resolution 640×480 as input to our network, train our model

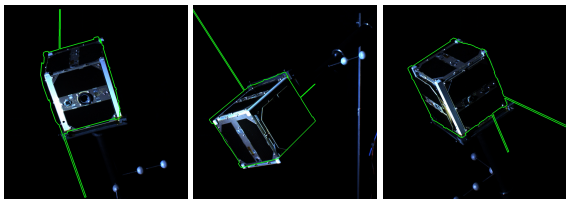


Figure 11: **Qualitative results on real data.** Our model easily adapts to real data, using as few as 20 annotated images.

	PVNet	SimplePnP	Hybrid	**Ours**
Ape	15.8	19.2	20.9	**22.3**
Can	63.3	65.1	75.3	**77.8**
Cat	16.7	18.9	24.9	**25.1**
Driller	65.7	69.0	70.2	**70.6**
Duck	25.2	25.3	27.9	**30.2**
Eggbox*	50.2	52.0	52.4	**52.5**
Glue*	49.6	51.4	53.8	**54.9**
Holepun.	39.7	45.6	54.2	**55.6**
Avg.	40.8	43.3	47.5	**48.6**

Table 5: **Comparison on Occluded-LINEMOD.** We compare our results with those of PVNet [31], SimplePnP [10] and Hybrid [36]. Symmetry objects are denoted with "*".

on the LINEMOD [8] dataset and test it on Occluded-LINEMOD without overlapped data. Although our framework supports multi-object training, for the evaluation to be fair, we train one model for each object type and compare it with methods not relying on another refinement procedure. Considering the small depth variations in this dataset, we remove the two pyramid levels with the largest reception fields from our framework, leaving only $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{F}_3$. As shown in Table 5, our model outperforms the state of the art even in this general 6D object pose estimation scenario.

5. Conclusion

We have proposed to use a single hierarchical network to estimate the 6D pose of an object subject to large scale variations, as would be the case in a space scenario. Our experiments have evidenced that training the different level of the resulting pyramid for different object scales and fusing their predictions during inference improves accuracy and robustness. We have also introduced the SwissCube dataset, the first satellite dataset with an accurate 3D model, physically-based rendering, and physical simulations of the Sun, the Earth, and the stars. Our approach outperforms the state of the art in both the wide-depth-range scenario and the more classical Occluded-LINEMOD dataset. In the future, we will concentrate on other important aspects of 6D object pose estimation in space, such as removing jitter by 6D pose tracking, and training a usable model with fully-unsupervised real data.

Acknowledgments. This work was supported by the Swiss Innovation Agency (Innosuisse). We would like to thank the EPFL Space Center (eSpace) for the data support.

References

[1] Adrien Bartoli, Vincent Gay-Bellile, Umberto Castellani, Julien Peyras, Søren Olsen, and Patrick Sayd. Coarse-to-Fine Low-Rank Structure-From-Motion. In *Conference on Computer Vision and Pattern Recognition*, 2008.

[2] Bo Chen, Jiewei Cao, Alvaro Parra, and Tat-Jun Chin. Satellite Pose Estimation with Deep Landmark Regression and Nonlinear Pose Refinement. In *International Conference on Computer Vision Workshops*, 2019.

[3] David Nash. The HYG database. http://www.astronexus.com/hyg, 2006.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Conference on Computer Vision and Pattern Recognition*, 2009.

[5] Simone D'Amico, Mathias Benn, and John L. Jørgensen. Pose estimation of an uncooperative spacecraft from actual space imagery. In *International Journal of Space Science and Engineering*, 2014.

[6] Ross Girshick. Fast R-CNN. In *International Conference on Computer Vision*, 2015.

[7] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[8] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes. In *Asian Conference on Computer Vision*, 2012.

[9] Tomáš Hodaň, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glent Buch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, Caner Sahin, Fabian Manhardt, Federico Tombari, Tae-Kyun Kim, Jiří Matas, and Carsten Rother. BOP: Benchmark for 6D Object Pose Estimation. In *European Conference on Computer Vision*, 2018.

[10] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-Stage 6D Object Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[11] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-Driven 6D Object Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[12] Yinlin Hu, Yunsong Li, Rui Song, Peng Rao, and Yangli Wang. Minimum Barrier Superpixel Segmentation. *Image and Vision Computing*, 70, 2018.

[13] Yinlin Hu, Rui Song, and Yunsong Li. Efficient Coarse-to-Fine PatchMatch for Large Displacement Optical Flow. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[14] Yinlin Hu, Rui Song, Yunsong Li, Peng Rao, and Yangli Wang. Highly Accurate Optical Flow Estimation on Superpixel Tree. *Image and Vision Computing*, 52, 2016.

[15] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial Transformer Networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[16] Omid Hosseini Jafari, Siva Karthik Mustikovela, Karl Pertsch, Eric Brachmann, and Carsten Rother. IPose: Instance-Aware 6D Pose Estimation of Partly Occluded Objects. In *Asian Conference on Computer Vision*, 2018.

[17] Longlong Jing, Yucheng Chen, and Yingli Tian. Coarse-To-Fine Semantic Segmentation from Image-Level Labels. *IEEE Transactions on Image Processing*, 29:225–236, 2019.

[18] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: Making Rgb-Based 3D Detection and 6D Pose Estimation Great Again. In *International Conference on Computer Vision*, 2017.

[19] Mate Kisantal, Sumant Sharma, Tae Ha Park, Dario Izzo, Marcus Märtens, and Simone D'Amico. Satellite Pose Estimation Challenge: Dataset, Competition Design and Results. In *IEEE Transactions on Aerospace and Electronic Systems*, 2020.

[20] Alexander Krull, Eric Brachmann, Frank Michel, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Learning Analysis-by-Synthesis for 6D Pose Estimation in RGB-D Images. In *International Conference on Computer Vision*, 2015.

[21] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. EPnP: An Accurate O(n) Solution to the PnP Problem. *International Journal of Computer Vision*, 2009.

[22] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep Iterative Matching for 6D Pose Estimation. In *European Conference on Computer Vision*, 2018.

[23] Zhigang Li, Gu Wang, and Xiangyang Ji. CDPN: Coordinates-Based Disentangled Pose Network for Real-Time Rgb-Based 6-DoF Object Pose Estimation. In *International Conference on Computer Vision*, 2019.

[24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In *International Conference on Computer Vision*, 2017.

[26] David G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 20(2):91–110, November 2004.

[27] Chien-Ping Lu, Gregory D. Hager, and Eric Mjolsness. Fast and Globally Convergent Pose Estimation from Video Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, 2000.

[28] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics*, 38(6):1–17, 2019.

[29] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making Deep Heatmaps Robust to Partial Occlusions for 3D Object Pose Estimation. In *European Conference on Computer Vision*, 2018.

[30] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-DoF Object Pose from Semantic Keypoints. In *International Conference on Robotics and Automation*, 2017.

[31] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. PVNet: Pixel-Wise Voting Network for 6DoF Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[32] Antoine Petit, Eric Marchand, and Keyvan Kanani. Vision-Based Space Autonomous Rendezvous: A Case Study. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[33] Mahdi Rad and Vincent Lepetit. BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects Without Using Depth. In *International Conference on Computer Vision*, 2017.

[34] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. In *arXiv Preprint*, 2018.

[35] Sumant Sharma, Jacopo Ventura, and Simone D'Amico. Robust Model-Based Monocular Pose Initialization for Noncooperative Spacecraft Rendezvous. In *Journal of Spacecraft and Rockets*, 2018.

[36] Chen Song, Jiaru Song, and Qixing Huang. HybridPose: 6D Object Pose Estimation Under Hybrid Representations. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[37] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[38] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully Convolutional One-Stage Object Detection. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[39] Engin Tola, Vincent Lepetit, and Pascal Fua. DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830, 2010.

[40] Tomasz Trzcinski, Christos Marios Christoudias, Vincent Lepetit, and Pascal Fua. Learning Image Descriptors with the Boosting-Trick. In *Advances in Neural Information Processing Systems*, December 2012.

[41] Shubham Tulsiani and Jitendra Malik. Viewpoints and Keypoints. In *Conference on Computer Vision and Pattern Recognition*, 2015.

[42] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *Robotics: Science and Systems Conference*, 2018.

[43] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. DPOD: 6D Pose Object Detector and Refiner. In *International Conference on Computer Vision*, 2019.

[44] Shijie Zhang and Xibin Cao. Closed-Form Solution of Monocular Vision-Based Relative Pose Determination for RVD Spacecrafts. In *Aircraft Engineering and Aerospace Technology*, 2005.

[45] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the Gap Between Anchor-Based and Anchor-Free Detection via Adaptive Training Sample Selection. In *Conference on Computer Vision and Pattern Recognition*, 2020.