Kaleidoscopic Background Attack: Disrupting Pose Estimation with Multi-Fold Radial Symmetry Textures

Supplementary Material

Contents

A . An Interpretation of Projected Orientation Consistency Loss Function 2
B . Multi-View Rendering with Diverse Augmentations 3
C . Experimental Parameter Settings in Digital and Physical Worlds 5
D . Experiments for Nature and KBA_{nat} in the Digital World 6
E . Experiments for Structure from Motion (SfM) Method 7
F . Experiments on Data and Scene Generalization 8
G . Experiments for Ablation Studies 9
H . Details on Clipping Pixel Colors to the CMYK Color Space 10
I . Details on Data for Experiments in the Digital and Physical Worlds 11
J . Details on Average Flow Direction Calculation Across Multiple Bisections 12
K . Additional Explanations for Adversarial Transferability 13
L . Additional Visualizations of 3D Reconstruction Results of DUSt3R 13
M. Additional Visualizations of Pose Estimation Results for KBA_{opt} in the Physical World 15

A. An Interpretation of Projected Orientation Consistency Loss Function

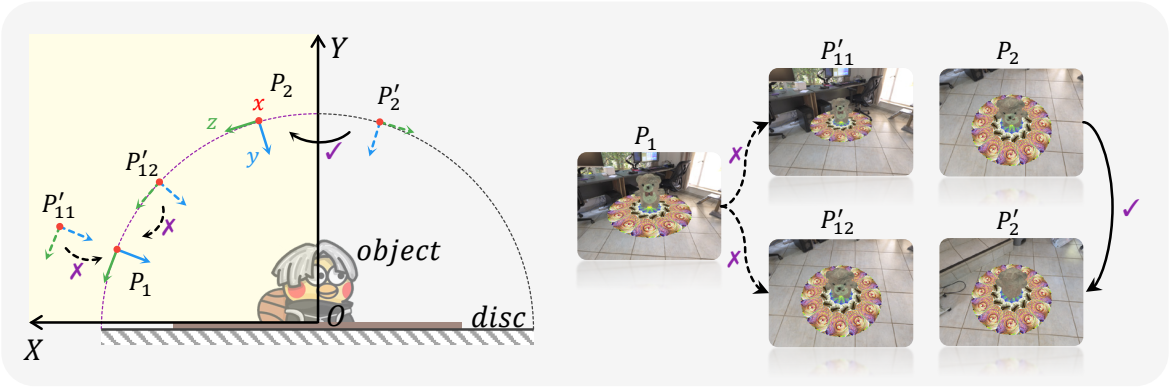


Figure 1. Illustration of different camera pose scenarios. Our KBA attack primarily enhances the projected orientation consistency by modifying the yaw angle between two camera poses, such as from P'_2 to P_2. For camera poses, such as P_1 and P'_{11}, or P_1 and P'_{12}, where there is minimal variation in the pitch angle or the distance to the center of the disc, our loss function $\mathcal{L}_{poc}$ has little effect on optimizing these poses. This is because the direction vectors of their coordinate axes (represented by the red, blue, and green arrows) have identical projection directions onto the disc.

The definition of our projected orientation consistency loss is given by Eq. 1. By assuming orthogonal projection and considering the symmetry of the disc, maximizing $\mathcal{L}_{poc}$ is approximately equivalent to maximizing the sum of cosine similarities between the projected camera orientation vectors $\hat{\mathbf{r}}_i$ for different poses, as shown in Eq. 2. In this section, we will analyze the special cases of camera poses that are not optimized, based on the projection of the camera coordinate axis onto the disc.

$$\mathcal{L}_{poc} = \sum_{i=1}^{3} \frac{\bar{\boldsymbol{\tau}}_i^a \cdot \bar{\boldsymbol{\tau}}_i^b}{\|\bar{\boldsymbol{\tau}}_i^a\| \|\bar{\boldsymbol{\tau}}_i^b\|} \tag{1}$$

$$\arg\max_{I_s} \mathcal{L}_{poc} \Leftrightarrow \arg\max_{I_s} \sum_{i=1}^{3} \frac{\hat{\mathbf{r}}_i^a \cdot \hat{\mathbf{r}}_i^b}{\|\hat{\mathbf{r}}_i^a\| \|\hat{\mathbf{r}}_i^b\|} \tag{2}$$

Intuitively, the optimization process described in Eq. 2 achieves its maximum when all vectors are aligned in the same direction. In other words, this occurs when the direction vectors of the camera coordinate axes for two different poses have identical projections onto the disc. In object-centric scenarios, where cameras from various spatial positions are primarily oriented towards the object, a notable case that satisfies the aforementioned conditions involves camera poses within the plane defined by the OX and OY rays, as illustrated by the light yellow region in Fig. 1. To further explain, let us consider the camera poses P_1, P'_{11}, and P'_{12} as examples. The primary difference between the camera poses P'_{11} and P_1 lies in their distances from the center of the disc, while the variation between P'_{12} and P_1 involves changes in the camera's pitch angle. The direction vectors of the camera coordinate axes for P_1, P'_{11}, and P'_{12} project in the same direction on the disc; for instance, the y-axis vector (indicated by the blue arrow) projects in the -X direction. As a result, these poses are not affected by the optimization process. In fact, all camera poses with the x-axis (indicated by the red arrow) perpendicular to the yellow plane remain unaffected by constraints. Additionally, camera poses derived from translating P_1 in space are not influenced by the loss function, as our attack does not constrain the translation component of the camera poses.

$$\mathcal{L}_{oc} = \sum_{i=1}^{3} \frac{\mathbf{r}_i^a \cdot \mathbf{r}_i^b}{\|\mathbf{r}_i^a\| \|\mathbf{r}_i^b\|} \tag{3}$$

For pose estimation methods that directly output the matrix R, an ideal attack approach can be achieved by maximizing the sum of cosine similarities between the corresponding vectors $\mathbf{r}_i$ from two different viewpoints, as detailed in Eq. 3. In fact, using the complete orientation information in $\mathcal{L}_{oc}$ to optimize the adversarial attack is not necessarily advantageous compared to using the relaxed version $\mathcal{L}_{poc}$. Consider the following cases, where variations exist between two camera poses

in terms of pitch angle, roll angle, and spatial position. In such cases, the object, such as the disc, captured by the cameras will exhibit noticeable deformations or changes in size within the image, as illustrated in Fig. 1. In contrast, when only the yaw angle of different camera poses changes, such as between P_2 and P_2', the shape of the disc remains nearly unchanged. Evidently, when considering the complete pose information of two cameras, including rotation and translation, optimizing the disc background texture must account not only for texture similarity under different yaw angles but also for additional constraints to handle changes in the shape and size of the disc in the image. In such cases, the optimization for adversarial attacks becomes more challenging due to the smaller and more physically implausible solution space.

B. Multi-View Rendering with Diverse Augmentations

In this section, we will present the various data augmentation methods mentioned in Section 2.3 of the main text, along with their specific parameter settings. In physical-world adversarial attacks, data augmentation is often an essential technique for enhancing the effectiveness of the attack and mitigating overfitting. During the training phase of our kaleidoscopic background attack, continuous input of diversified images from two different perspectives is required to optimize the segmentation output. Therefore, we apply data augmentation to various resources and intermediate results within the rendering pipeline, as shown in Fig. 2. The following part details the data augmentation methods and the parameter settings used in this work.

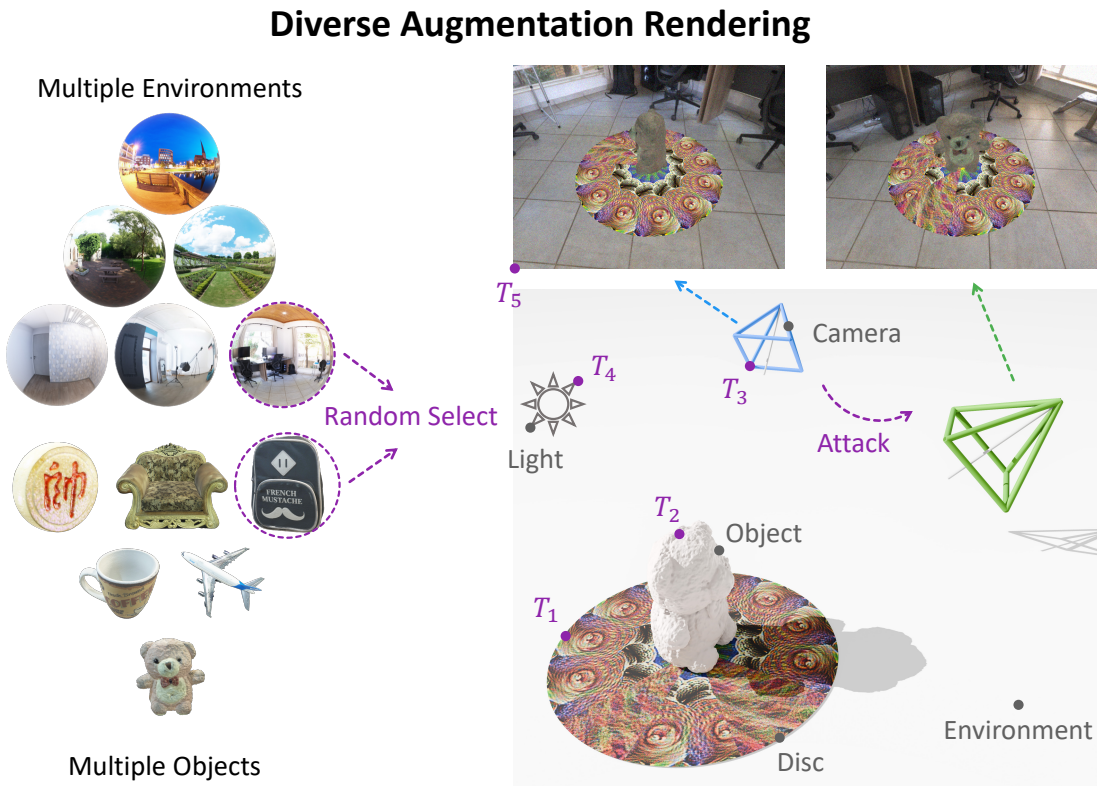


Figure 2. Illustration of diverse augmentation rendering techniques utilized in the KBA_{opt} method to optimize kaleidoscopic segments. Each rendering process generates images from two distinct viewpoints. For each instance, an object and an environment are randomly selected from predefined collections. Various augmentations are applied to enhance data variability: **T1**, where applying random downsampling or upsampling to the texture images; **T2**, where objects undergo random rotations and scaling; **T3**, where pitch, yaw, roll, distance, and viewpoints of the two randomly positioned cameras are each randomly adjusted; **T4**, where ambient lighting is modified by random adjustments in color, brightness, and position; and **T5**, where Gaussian noise is added to the rendered images to further introduce variability.

$\mathbf{T_1}$: To enhance the effectiveness of printed disc textures at different scales of attack, inspired by the concept of mipmap in computer graphics, we apply random down-sampling and up-sampling to the texture mapping. Specifically, we first down-sample the texture map to a random scale between 0.5 and 1 times its original size, and then use bilinear interpolation to scale it back to its original size. This approach reduces the dependency of adversarial attacks on excessively high-frequency

information, making the optimized texture more stable. The effect of down-sampling and up-sampling is similar to the total variation (TV) loss function commonly used in physical-world adversarial attacks. However, unlike TV loss, this method avoids adding extra loss functions that could disrupt the stability of target optimization. For the mesh, each rendering involves a random rotation around the symmetrical axis of the disc, with an angle ranging from 0 to 360 degrees.

T_2: For each rendering, we randomly select an object from the set of 3D objects used for training, apply a random scaling, and ensure that the maximum dimension after scaling falls within a random size range of 0.4 to 1.0 meters. In the RDF coordinate system setup, the object can rotate around the x-axis or z-axis by 0 to 180 degrees, and around the y-axis by 0 to 360 degrees. In this context, although the object may be positioned on the plane in an unphysical manner, it provides sufficient object diversity to enhance our optimization. For the position of the object, we place it on the surface of the disc at a random location within a circle of radius 0.1 meters centered at the center of the disc.

T_3: For camera data augmentation, we randomly select the pose of one viewpoint and derive the pose of the second camera based on the first. According to the definition in the LookAt format, the camera pose is determined by the position it looks at, the rotation angle around the y-axis (i.e., yaw angle θ_y), the angle between the x-z plane (i.e., pitch angle θ_p), the distance d to the look-at position, and the up vector. The first camera looks at a random position within a circular region of radius 0.2 meters centered on the disc plane. The distance d is randomly selected from the range of 2.5 to 3.0 meters, the yaw angle θ_y is randomly chosen within the range of 0 to 360 degrees, and the pitch angle θ_p is randomly chosen from the range of 15 to 85 degrees. For the second camera, the distance d is slightly increased or decreased by 0.1 meters based on the first camera, the pitch angle θ_p is adjusted by a small increment or decrement of 10 degrees, and the yaw angle θ_y is randomly chosen within the range of 0 to 360 degrees, while the remaining parameters are kept consistent with the first camera.

T_4: During the data augmentation process, we use a point light source to simulate illumination. The x-coordinate of the light source position is randomly chosen from -10 to 10 meters, and the y-coordinate is randomly chosen from -10 to 0 meters, and the z-coordinate is randomly chosen from -10 to 10 meters. The ambient color of the light source is randomly chosen within the range of 0.6 to 1.0, the diffuse color is randomly chosen within the range of 0 to 0.1, and the specular color is randomly chosen within the range of 0 to 0.1. For each of the three channels of ambient color, diffuse color, and specular color, the value is adjusted by ± 0.1 to simulate various color temperatures of the lighting.

T_5: For the rendered images from the two viewpoints, we add Gaussian noise to simulate signal noise encountered when taking indoor photographs. The Gaussian noise has a mean of 0 and a variance of 5.

For each scene's background, we randomly select an environment from a set for each rendering. HDRI images are mapped to the inner surface of a sphere with a large radius of $10,000$ meters. While this method does not produce realistic physical changes when camera poses are altered, it provides a simple and effective way to achieve varied background information. In contrast, using complex meshes for background rendering can offer accurate perspective changes but significantly increase rendering time, adding substantial computational cost to each step of adversarial optimization.

In each rendering process, we output two images with different viewpoints. Apart from the camera pose-related augmentation T_3, other data augmentations are applied to the overall scene setup and are used only once per rendering. In other words, for the two images rendered from different viewpoints, the corresponding scene information, including the background disc, objects, lighting, and other background elements, remains consistent. In this work, we utilized nearly all conceivable data augmentation methods applicable during the rendering process. Considering that adjusting the parameters of a specific data augmentation method has a negligible impact on the overall data diversity, we did not conduct additional experiments to explore the effect of such a large number of data augmentation methods on the experimental results.

C. Experimental Parameter Settings in Digital and Physical Worlds

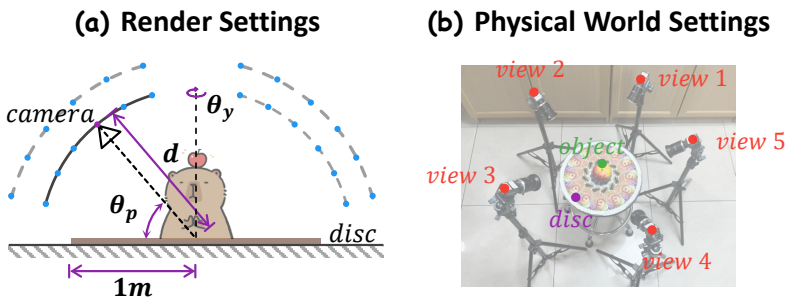


Figure 3. (a) The setup for rendering scenes in the digital world. (b) The setup for testing scenes in the physical world.

Training datasets and parameters. We utilized six HDRI images, including three indoor and three outdoor scenes from the Polyhaven [4] website. These images are mapped onto a spherical mesh to create a realistic background. For the 3D objects, we selected 32 items spanning 20 categories from the OmniObject3D [15] dataset. Each object was placed at the center of a disc with a radius of 1 m. Multiple data augmentations were applied to the disc texture, 3D objects, lighting conditions, camera poses, and rendered images to increase data diversity during the optimization. In this paper, we use a desktop texture as the natural texture, termed **Nature** in the experiments. For the kaleidoscopic series of adversarial attacks, we created a disc with $N = 12$ segments, each spanning $\theta = 30°$. The approach that uses natural textures as segments is referred to as $\mathbf{KBA_{nat}}$. In contrast, $\mathbf{KBA_{opt}}$ refers to the optimization of these segments using the proposed projected consistency loss. The optimization is conducted for about 20k steps, with segment colors constrained to the CMYK color space every 500 steps.

Test setup for the digital world. In this phase, we collected 10 HDRI images from Polyhaven and 25 objects across 25 categories from the OmniObject3D dataset for evaluation. The test environments and objects do not overlap with those used during training. The rendered scene comprises the 3D object, background disc, and environment, as shown in Fig. 3(a). Similar to the rendered scene during optimization, the disc radius is set to 1 m, with the objects placed at the center surface of the disc and scaled to fit within a $0.8 \times 0.8 \times 0.8$ m bounding box. The rendering camera faces the disc center at a distance d, with pitch angle θ_p defined between the line to the camera and the disc plane, and yaw angle θ_y indicating rotation around the disc's symmetry axis. In order to increase the diversity of experiments and the generalization of results, we configured $6 \times 36 \times 6 = 1296$ parameter combinations using six pitch angles (ranging from $10°$ to $85°$), 36 yaw angles (in increments of $10°$), and six distances (ranging from 2.0 m to 3.0 m). Images and masks for the 3D objects, discs, and environments were rendered and combined during testing to produce the final image. We designed two testing scenarios, **DT1** and **DT2**, to assess the impact of different background discs on camera pose estimation. In **DT1**, the pitch angle is fixed at $55°$ and the distance at 2.4 m, while varying the yaw angles. In **DT2**, 1296 camera poses are randomly selected for a more comprehensive evaluation. Each scenario includes four samples per object-environment combination, resulting in a total of $25 \times 10 \times 4 = 1000$ samples.

Test setup for the physical world. We selected 24 models as 3D objects, including vegetables, fruits, animals, and vehicles. To ensure complete capture of the disc while maintaining good imaging quality for the objects, we crafted two discs: one with a radius of 15 cm for objects ranging from 10 to 20 cm, and another with a radius of 20 cm for objects ranging from 20 to 30 cm. As shown in Fig. 3(b), five industrial cameras are evenly distributed around the disc, with their lenses directed toward the center at distances ranging from 20 to 50 cm. Under normal indoor lighting conditions, we calibrated the cameras using a calibration board and simultaneously captured object-centric images from five different viewpoints. We captured five groups of images for each object, adjusting the camera poses between groups to ensure data diversity.

Evaluation metircs. Following [5, 6, 13, 14, 16], we evaluated the accuracy of pose estimation using Relative Rotation Accuracy (RRA), Relative Translation Accuracy (RTA), and mean Average Accuracy (mAA). Specifically, RRA compares the relative rotation $R_i R_j^\top$ from the i-th to the j-th camera with the ground truth $R_i^\star R_j^{\star\top}$, while RTA measures the angle between the predicted vector T_{ij} and the ground truth vector $T_{ij}^\star$ pointing from camera i to camera j. We report RTA@γ and RRA@γ ($\gamma \in \{5, 15, 30\}$), representing the percentage of camera pairs with RRA or RTA values below the threshold γ. Furthermore, we compute the mAA(30), which is defined as the area under the accuracy curve of angular differences at $min(\text{RRA@30}, \text{RTA@30})$. Beyond these three standard metrics, we introduce a custom Relative Rotation Similarity (RRS) metric, leveraging cosine similarity to assess the similarity between different predicted relative rotations $R_i R_j^\top$. An RRS value close to 1 signifies high consistency in pose orientations.

D. Experiments for Nature and KBA_{nat} in the Digital World

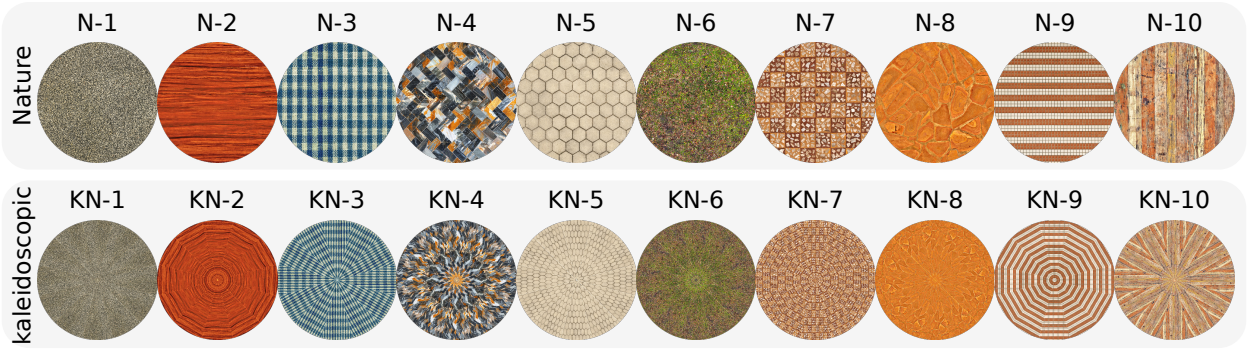


Figure 4. Various nature textures and their corresponding images with kaleidoscopic patterns. Specifically: **N-1** is the bicolour gravel texture scanned from a road surface; **N-2** is the dark wood texture; **N-3** is the fabric pattern texture; **N-4** is the grey cartago texture scanned from ceramic tiles; **N-5** is the hexagonal concrete paving texture; **N-6** is the leafy grass texture obtained by scanning a lawn; **N-7** is the marble mosaic tiles texture; **N-8** is a slate floor texture; **N-9** is a square tiled wall texture; and **N-10** is the worn planks texture scanned from an outdoor tabletop.

Textures	RRA@5 ↑	RRA@15 ↑	RRA@30 ↑	RTA@5 ↑	RTA@15 ↑	RTA@30 ↑	mAA(30) ↑	RRS
N-1	0.83 ±0.04	0.98 ±0.01	0.98 ±0.01	0.56 ±0.08	0.88 ±0.06	0.96 ±0.02	0.76 ±0.05	0.64 ±0.01
N-2	0.60 ±0.08	0.98 ±0.01	**1.00** ±0.00	0.47 ±0.06	0.87 ±0.06	0.96 ±0.03	0.71 ±0.04	0.64 ±0.01
N-3	0.51 ±0.09	0.89 ±0.02	0.94 ±0.01	0.40 ±0.05	0.78 ±0.05	0.90 ±0.02	0.63 ±0.02	0.64 ±0.01
N-4	0.77 ±0.03	0.93 ±0.01	0.96 ±0.01	0.48 ±0.10	0.77 ±0.08	0.89 ±0.04	0.67 ±0.07	0.64 ±0.01
N-5	0.67 ±0.05	0.91 ±0.02	0.95 ±0.02	0.43 ±0.07	0.78 ±0.07	0.90 ±0.04	0.66 ±0.05	0.64 ±0.01
N-6	0.90 ±0.03	**0.99** ±0.00	0.99 ±0.00	**0.65** ±0.08	**0.91** ±0.05	**0.97** ±0.02	**0.81** ±0.04	0.64 ±0.01
N-7	**0.92** ±0.02	0.97 ±0.01	0.98 ±0.01	0.60 ±0.10	0.87 ±0.06	0.95 ±0.02	0.78 ±0.05	0.64 ±0.01
N-8	0.81 ±0.04	0.98 ±0.01	0.99 ±0.00	0.49 ±0.09	0.82 ±0.07	0.94 ±0.04	0.71 ±0.06	0.64 ±0.01
N-9	0.53 ±0.06	0.82 ±0.03	0.88 ±0.02	0.36 ±0.06	0.68 ±0.06	0.81 ±0.03	0.57 ±0.04	0.63 ±0.01
N-10	0.79 ±0.04	0.97 ±0.01	0.98 ±0.01	0.48 ±0.09	0.83 ±0.07	0.93 ±0.04	0.71 ±0.05	0.64 ±0.01

Table 1. Experimental results of pose estimation using the DUSt3R model on various nature textures in the digital world. Each cell contains two values: the larger value represents the mean of the metric across all samples, while the smaller value indicates the standard deviation of the metric across different object categories. Bold values indicate the best performance for pose estimation. N-10 corresponds to Nature as referenced in the main text.

As shown in Fig. 4, we selected a variety of textures commonly seen in daily life. Using these textures, we created nature discs and background discs with 12-fold kaleidoscopic patterns to investigate their impact on the camera pose estimation results of the DUSt3R model.

Similar to the setup in the discussion section of the main text, we conducted experiments on 1,000 groups of 5-view samples with a fixed distance of 2.4 meters and a pitch angle of 55 degrees, while the yaw angle was randomly selected between 0 and 360 degrees. The experimental results using discs with nature textures are shown in Tab. 2. Based on metrics such as RRA@15, RRA@30, RTA@15, and RTA@30, it can be observed that most natural textures yield relatively good results for camera pose estimation. However, when evaluated using stricter metrics such as RRA@5, RTA@5, and mAA(30), significant differences can still be observed between different textures. The N-3 and N-9 textures, which have distinct vertical stripe patterns, show relatively poorer results in camera pose estimation. One possible reason is that these pronounced vertical stripe patterns can be approximated as axisymmetric designs, making it difficult to distinguish between certain viewpoints, thereby impacting the accuracy of camera pose estimation. From the experimental results, we found that the camera pose estimation results for N-6 and N-1 were relatively better. Interestingly, for N-6 and particularly N-1, the pixel details are so fine that, from a human perspective, the differences between various directions are not apparent. However, upon closer inspection, textures like N-6 show slight variations, with the upper right appearing slightly greener and the lower left slightly

browner, creating some directional differences. These findings suggest that if we aim to improve camera pose estimation by optimizing background textures, one potential solution is to introduce directional variations within fine textures.

Textures	RRA@5 ↓	RRA@15 ↓	RRA@30 ↓	RTA@5 ↓	RTA@15 ↓	RTA@30 ↓	mAA(30) ↓	RRS ↑
KN-1	0.30 ±0.06	0.39 ±0.07	0.48 ±0.08	0.17 ±0.02	0.35 ±0.02	0.48 ±0.04	0.28 ±0.02	0.71 ±0.01
KN-2	0.27 ±0.06	0.50 ±0.06	0.62 ±0.05	0.16 ±0.03	0.42 ±0.03	0.57 ±0.03	0.32 ±0.02	0.64 ±0.01
KN-3	0.35 ±0.08	0.70 ±0.04	0.83 ±0.03	0.24 ±0.03	0.58 ±0.05	0.74 ±0.04	0.45 ±0.02	0.64 ±0.01
KN-4	0.17 ±0.07	0.31 ±0.08	0.40 ±0.09	0.03 ±0.01	0.14 ±0.02	0.27 ±0.03	0.10 ±0.01	0.78 ±0.01
KN-5	0.38 ±0.05	0.56 ±0.05	0.64 ±0.05	0.21 ±0.03	0.47 ±0.04	0.61 ±0.03	0.38 ±0.03	0.66 ±0.01
KN-6	0.27 ±0.05	0.34 ±0.06	0.43 ±0.07	0.15 ±0.02	0.30 ±0.02	0.43 ±0.03	0.24 ±0.02	0.72 ±0.01
KN-7	0.34 ±0.05	0.45 ±0.04	0.52 ±0.05	0.20 ±0.03	0.40 ±0.03	0.51 ±0.03	0.32 ±0.02	0.67 ±0.01
KN-8	0.30 ±0.06	0.48 ±0.07	0.57 ±0.06	0.16 ±0.02	0.38 ±0.03	0.52 ±0.03	0.30 ±0.02	0.68 ±0.01
KN-9	0.34 ±0.05	0.64 ±0.03	0.76 ±0.03	0.23 ±0.04	0.50 ±0.05	0.66 ±0.05	0.39 ±0.03	0.64 ±0.01
KN-10	0.27 ±0.06	0.46 ±0.06	0.55 ±0.06	0.15 ±0.02	0.38 ±0.03	0.53 ±0.03	0.29 ±0.02	0.67 ±0.01
KBA_{opt}	**0.13** ±0.06	**0.22** ±0.07	**0.29** ±0.10	**0.02** ±0.01	**0.11** ±0.02	**0.24** ±0.04	**0.08** ±0.02	**0.86** ±0.01

Table 2. Experimental results of pose estimation using the DUSt3R model on various discs constructed from different natural kaleidoscope segments in the digital world. Each cell contains two values: the larger value represents the mean of the metric across all samples, while the smaller value indicates the standard deviation of the metric across different object categories. Bold values indicate the best performance against adversarial attacks, while underlined values represent the second-best results. KN-10 corresponds to KBA_{nat} as referenced in the main text.

The experimental results for the background discs with 12-fold kaleidoscopic patterns are shown in Tab. 2. From the experimental results, we found no clear correlation between the effectiveness of adversarial attacks using natural segment images and the quality of camera pose estimation results with those same natural textures. The adversarial attack results for KN-4 were significantly better than those for other textures. Referring to Fig. 4, one possible explanation is that the high-frequency details in the KN-4 texture are more pronounced than in the other textures, which enhances the interference effect of repeating symmetrical patterns across different viewpoints. From the experimental results, it can be observed that the adversarial attack performance of the background discs composed of the 10 natural textures as segment images is inferior to that of our optimization-based KBA_{opt} method. This further demonstrates the effectiveness of our optimization process and the projected orientation consistency loss.

E. Experiments for Structure from Motion (SfM) Method

Methods	Textures	RRA@30 ↓	RTA@30 ↓	mAA(30) ↓	RRS ↑	VR (%)
COLMAP+SPSG	Nature	0.01 ±0.02	0.16 ±0.10	**0.00** ±0.00	0.59 ±0.07	58.50 ±9.08
	KBA_{nat}	**0.00** ±0.00	0.15 ±0.05	**0.00** ±0.00	0.86 ±0.03	98.67 ±4.57
	KBA_{opt}	**0.00** ±0.01	**0.11** ±0.04	**0.00** ±0.00	**0.90** ±0.03	97.33 ±5.62

Table 3. Experimental results of COLMAP+SPSG using various background discs in a physical environment. Each cell contains two values: the larger value represents the mean of the metric across all samples, while the smaller value indicates the standard deviation of the metric across different object categories. Bold values indicate the best performance against adversarial attacks.

Although SfM techniques are not ideally suited for camera pose estimation in sparse image tasks, we selected the COLMAP+SPSG method, which builds on the popular SfM pipeline COLMAP [12] enhanced with the SuperPoints [2] and SuperGlue [11], to evaluate the effectiveness of adversarial attacks, as shown in Tab. 3. The experimental results show that for textures with natural patterns, the ratio of valid pose estimations (VR) is less than 60% of the total number of images. This suggests that, due to the sparse image setup, accurately estimating most viewpoints is challenging. Therefore, we analyzed the RRA, RTA, mAA, and RRS metrics for all successfully estimated views. Based on the RTA and RRS metrics, our KBA method demonstrates a certain level of adversarial effectiveness, with the optimized KBA_{opt} method slightly outperforming the KBA_{nat} method that uses natural textures as segment images. Notably, the VR metric for both KBA_{nat}

and KBA_{opt} is significantly higher compared to the Nature method. This indicates that our radially symmetric kaleidoscopic pattern textures can, to some extent, lead the model to make ”confident” but erroneous camera pose estimations.

F. Experiments on Data and Scene Generalization

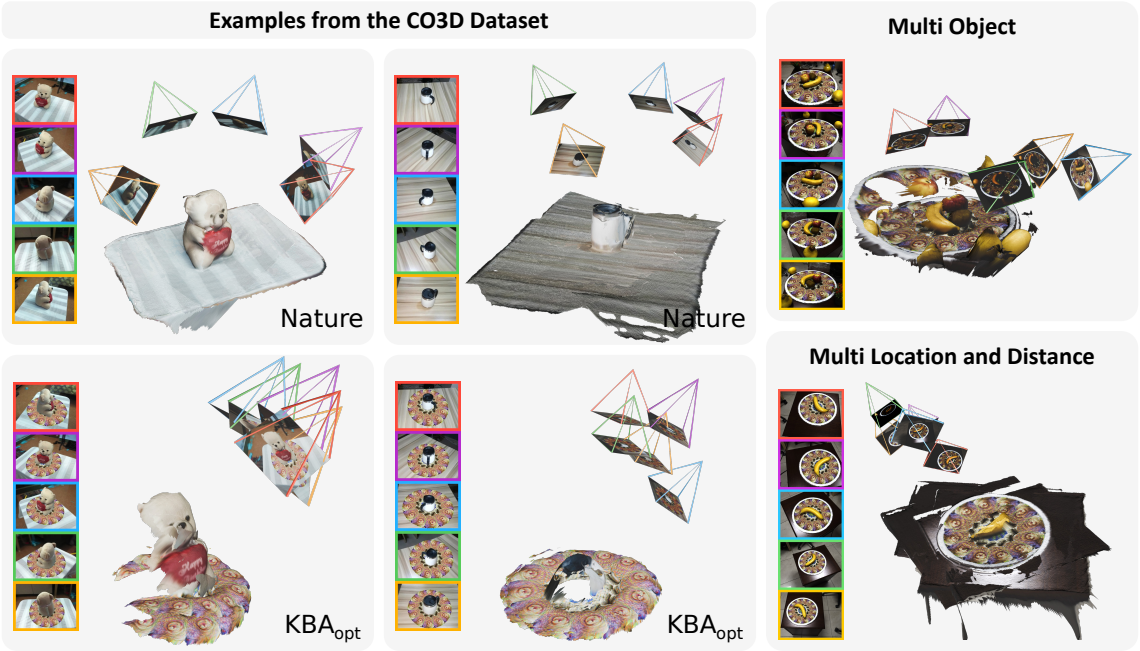


Figure 5. DUSt3R results on two sample sets from the CO3D dataset before and after incorporating adversarial background disks optimized by KBA_{opt}, along with adversarial attack results under multi-object scenarios, various disk positions, and different camera distances.

For the task of sparse-view camera pose estimation, it is typically assumed that an object is located at the center of the scene, with cameras distributed around the object for capturing images. Consequently, in this work, our scene is configured with an object placed on a background disc. We validate the effectiveness of our method on both digitally rendered data and physically captured images. To further investigate the adversarial performance beyond the training data, in this section, we evaluate the camera pose estimation results before and after adversarial attacks on the CO3D dataset [8], as well as the adversarial performance in multi-object scenes and under complex viewpoints, as shown in Fig. 5. We use Photoshop to add the optimized adversarial background textures beneath the objects in the original images from the CO3D dataset. This experiment, situated between the digital and physical worlds, further validates the effectiveness of our adversarial attack.

For scenarios where multiple objects are scattered near the table and on the disk, the adversarial textures optimized by KBA_{opt} can also effectively disrupt the model's predictions. When the distances from the scene center are inconsistent and the disc is located at different positions within the image, our adversarial attack still demonstrates strong generalizability.

G. Experiments for Ablation Studies

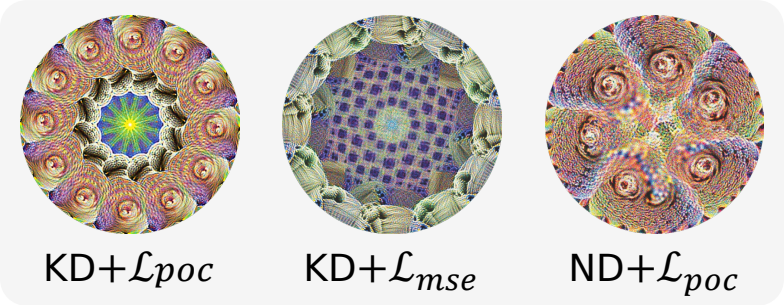


Figure 6. Textures of background discs generated from various method combinations. KD denotes the initialization and assembly of the disc using segments with kaleidoscopic patterns, while ND refers to direct initialization with a complete circular pattern. $\mathcal{L}_{poc}$ represents the projected orientation consistency loss used in our main text, and $\mathcal{L}_{mse}$ denotes the mean square error loss function. KD + $\mathcal{L}_{poc}$ corresponds to the KBA_{opt} method discussed in our main text, KD + $\mathcal{L}_{mse}$ is included to investigate the impact of the loss function on the effectiveness of adversarial attacks, and ND + $\mathcal{L}_{poc}$ is used to study the effect of the kaleidoscopic background disc construction method on the effectiveness of adversarial attacks.

In this section, we will explore the effectiveness of the kaleidoscopic background and the projected orientation consistency loss. We denote the use of the kaleidoscopic background disc as KD, while using a plain circular disc is denoted as ND. $\mathcal{L}_{poc}$ refers to the projected orientation consistency loss used in this work, as shown in Eq. 4.

$$\mathcal{L}_{poc} = \sum_{i=1}^{3} \frac{\bar{\boldsymbol{\tau}}_i^a \cdot \bar{\boldsymbol{\tau}}_i^b}{\|\bar{\boldsymbol{\tau}}_i^a\| \|\bar{\boldsymbol{\tau}}_i^b\|} \tag{4}$$

As a comparison to $\mathcal{L}_{poc}$, we employ a straightforward adversarial attack method, which minimizes the mean square error between the DUSt3R output pointmaps $O^a \in \mathbb{R}^{H \times W \times 3}$ from view a and $O^b \in \mathbb{R}^{H \times W \times 3}$ from view b to achieve adversarial attacks. The definition of this loss function $\mathcal{L}_{mse}$ is shown in Eq. 5, where o_k^a and o_k^b represent the k-th element of O^a and O^b, respectively.

$$\mathcal{L}_{mse} = \frac{1}{K} \sum_{k=1}^{K} \left(o_k^a - o_k^b\right)^2, \; K = H \times W \times 3 \tag{5}$$

Methods	RRA@5 ↓	RRA@15 ↓	RRA@30 ↓	RTA@5 ↓	RTA@15 ↓	RTA@30 ↓	mAA(30) ↓	RRS ↑
KD + $\mathcal{L}_{\text{poc}}$	**0.13** ±0.06	**0.22** ±0.07	**0.29** ±0.10	**0.02** ±0.01	**0.11** ±0.02	**0.24** ±0.04	**0.08** ±0.02	**0.86** ±0.01
KD + $\mathcal{L}_{\text{mse}}$	0.14 ±0.09	0.27 ±0.12	0.29 ±0.12	0.05 ±0.02	0.16 ±0.04	0.26 ±0.07	0.13 ±0.04	0.76 ±0.03
ND + $\mathcal{L}_{\text{poc}}$	0.35 ±0.13	0.59 ±0.08	0.62 ±0.08	0.12 ±0.01	0.35 ±0.02	0.51 ±0.03	0.29 ±0.02	0.70 ±0.01

Table 4. Experimental results of different disc construction methods combined with various loss functions. Each cell contains two values: the larger value represents the mean of the metric across all samples, while the smaller value indicates the standard deviation of the metric across different object categories. Bold values indicate the best performance against adversarial attacks.

From the experimental results shown in Tab. 4, it can be observed that KD + $\mathcal{L}_{poc}$, which corresponds to the KBA_{opt} method described in the main text, achieves the best adversarial attack performance. Using $\mathcal{L}_{mse}$ instead of $\mathcal{L}_{poc}$ to optimize the adversarial attack results in decreased effectiveness. This is because aligning points between the pointmaps in camera coordinates is not necessary for attacking the camera's pose orientation and can complicate the optimization process. Directly using a plain disc for optimization also fails to achieve effective adversarial attack results since relying solely on the loss function makes it difficult to produce satisfactory symmetrical effects. An interesting observation is that even when using a plain circular disc, our projected orientation consistency loss $\mathcal{L}_{poc}$ can still optimize the texture to form an approximate 6-fold radial symmetry, as shown in Fig. 6. This indicates that our loss function is well-matched with the construction of the kaleidoscopic background, and the solution for adversarial attacks on camera orientation tends to converge toward radial symmetry patterns.

H. Details on Clipping Pixel Colors to the CMYK Color Space

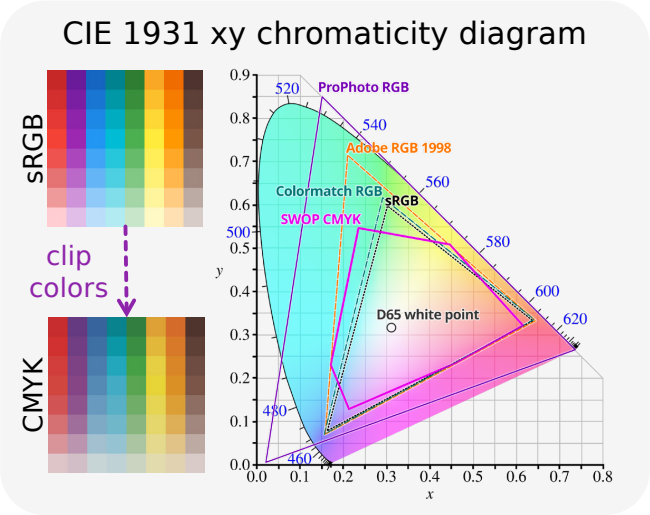


Figure 7. Illustration of images in the sRGB color space and their clipped versions in CMYK, along with a depiction of the relative positions of different color spaces on the CIE 1931 xy chromaticity diagram.

In this section, we present the technical details employed for pixel color clipping, as outlined in Section 2.3 of the main text.

To avoid impacting the optimization of adversarial attacks, we did not incorporate the Non-Printability Score (NPS) [3] into the loss function. Instead, to enhance the physical realizability of the background disc in a straightforward manner, we clipped the color of the texture image from the sRGB color space to the CMYK color space every 500 optimization steps. Specifically, we used the U.S. Web Coated (SWOP) [1] profile as the CMYK ICC configuration and employed the open-source Little-CMS [7] for color management. The process involved converting the texture image to a TIFF format and performing soft-proofing to simulate how the image would appear when printed on coated paper. During this step, an absolute colorimetric rendering intent was used to map colors precisely between the input and output while maintaining the white point. Finally, the soft-proofed image was reloaded as the updated, clipped texture image. In our work, we found that the simple color space clipping process described above, combined with modern, widely used art printing techniques, is sufficient to produce adversarial background discs that are nearly indistinguishable between the digital and physical worlds, thereby effectively enabling physical-world adversarial attacks.

I. Details on Data for Experiments in the Digital and Physical Worlds

Figure 8. Visualization of the environments and 3D objects used in this work for optimizing (training) against kaleidoscopic backgrounds, and for testing in both the digital and physical worlds.

In this section, we will provide the visualization results of all the environments, 3D objects, and physically captured image data used in this work. For the training data, we used six HDRI images, three from indoor scenes and three from outdoor scenes, sourced from the Polyhaven [4] website, as shown in Fig 8. These images were mapped onto a spherical mesh to create realistic backgrounds. For the 3D objects, we selected 32 items from 20 categories of the OmniObject3D [15] dataset. Each object was positioned at the center of a disc with a radius of 1 meter, ensuring consistent placement across all scenes. For the digital-world test data, we additionally collected 10 HDRI images from Polyhaven and 25 objects from 25 categories of the OmniObject3D dataset for evaluation. For the physical-world test data, we selected 24 models as 3D objects, including vegetables, fruits, animals, and vehicles.

J. Details on Average Flow Direction Calculation Across Multiple Bisections

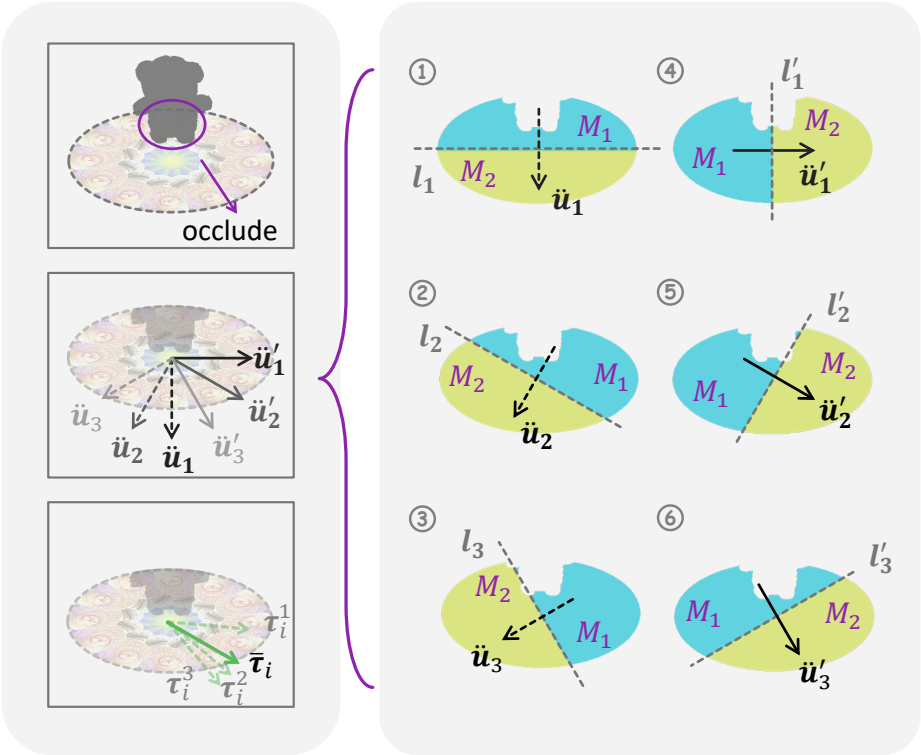


Figure 9. Illustration of the relationships between the three lines l, l', the sets M_1, M_2, and the vectors $\ddot{\mathbf{u}}$, $\ddot{\mathbf{u}}'$ after dividing the disc with three straight lines l.

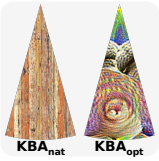
In this section, we will detail the method for calculating average coordinate flow directions in Section 2.2 of the main text.

To account for potential occlusions on the disc, we used three different lines inside the disc, l_j ($j = 1, 2, 3$), to estimate coordinate variations from multiple directions. As shown in Fig. 9, given a line l_j, we can define a corresponding line l'_j, where the two lines divide the disc into four semicircles. For each semicircle, we compute the operation defined in Eq. 6. To reduce redundant computations, it can be deduced that the angle between l_j and l_{j+1} should be set to 30 degrees when using three lines l_1, l_2, and l_3.

$$\frac{\sum_{m_k \in M_k} \Phi_i(m_k)}{|M_k|}, \; k = 1, 2 \tag{6}$$

Since most of the computations can be accelerated through parallel processing, using three lines for partitioning does not introduce significant additional computational overhead compared to using a single line. Additionally, the KBA_{opt} method using three lines has demonstrated its effectiveness through experiments. Therefore, we leave the exploration of the impact of the number of lines used for partitioning the disc on the adversarial attack for future work.

K. Additional Explanations for Adversarial Transferability



Method	SF	IE	HFR@50
KBA$_{nat}$	7.33	4.54	0.16%
KBA$_{opt}$	10.15	4.92	3.34%

Table 5. Visualization of kaleidoscopic segment images for KBA$_{nat}$ and KBA$_{opt}$ used in this study, along with their respective texture metrics. Here, SF denotes the spatial frequency of the image, IE represents the information entropy, and HFR@50 indicates the ratio of spectral energy within the central 50% of the Fourier spectrum to the total energy of the image. The optimized kaleidoscopic segments contain higher high-frequency information and greater informational content compared to natural textures.

The experimental results demonstrate that adversarial attacks using kaleidoscopic backgrounds formed from optimized segment images outperform those using nature texture segment images, not only in the DUSt3R model but also across other camera pose estimation models. The reason for this result can be attributed to two factors: firstly, the adversarial transferability commonly observed across deep learning models, where adversarial examples optimized on one model retain their disruptive effects when transferred to other models performing the same task. On the other hand, referring to the experimental results in Sec. D, we believe that the segment images optimized using our KBA$_{opt}$ method contain more high-frequency information compared to KBA$_{nat}$. This allows for easier matching of background textures across different viewpoints, both on the surrogate model and target models, thereby facilitating the adversarial attack on camera pose estimation.

As detailed in Tab. 5, we computed the spatial frequency (SF) [10], information entropy (IE) [9], and the ratio of spectral energy within the central 50% of the Fourier spectrum to the total energy of the image (HFR@50) for the segment images corresponding to KBA$_{nat}$ and KBA$_{opt}$. The experimental results show that the optimized segment image indeed contains more high-frequency information.

L. Additional Visualizations of 3D Reconstruction Results of DUSt3R

We visualized the 3D reconstruction and camera pose estimation results on DUSt3R for several sets of images, both before and after the attack, including results from digital-world rendering and physical-world capture. The visualizations are shown in Fig. 10, Fig. 11, and Fig. 12. From the experimental results, it can be observed that, DUSt3R is able to accurately estimate the camera pose for each image with a natural background disc and successfully perform 3D reconstruction of the objects. For images with a kaleidoscopic background disc, the camera pose orientations are attacked to nearly identical directions. As a result, the model's 3D point cloud predictions for each image overlap, resembling multiple layers of cabbage. Furthermore, as seen in the first image of the main text with the banana object, there is almost no object reconstruction on the kaleidoscopic background disc. This suggests that the DUSt3R model tends to treat the banana object as random noise from multiple images taken from the same viewpoint, thus ignoring its reconstruction.

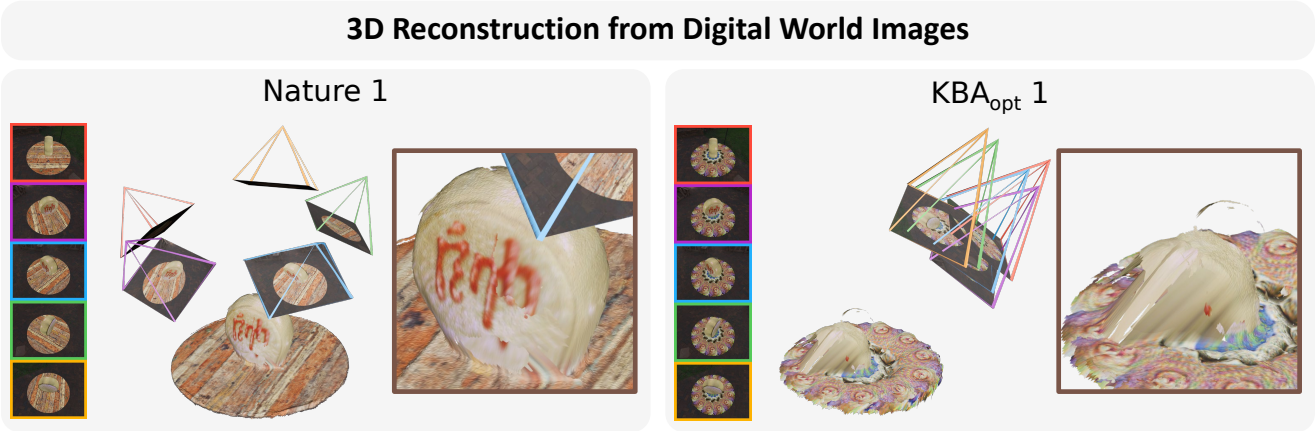


Figure 10. Visualization of 3D reconstruction and pose estimation results on the DUSt3R model for digital-world images of different objects and environments.

Figure 11. Visualization of 3D reconstruction and pose estimation results on the DUSt3R model for digital-world images of different objects and environments.

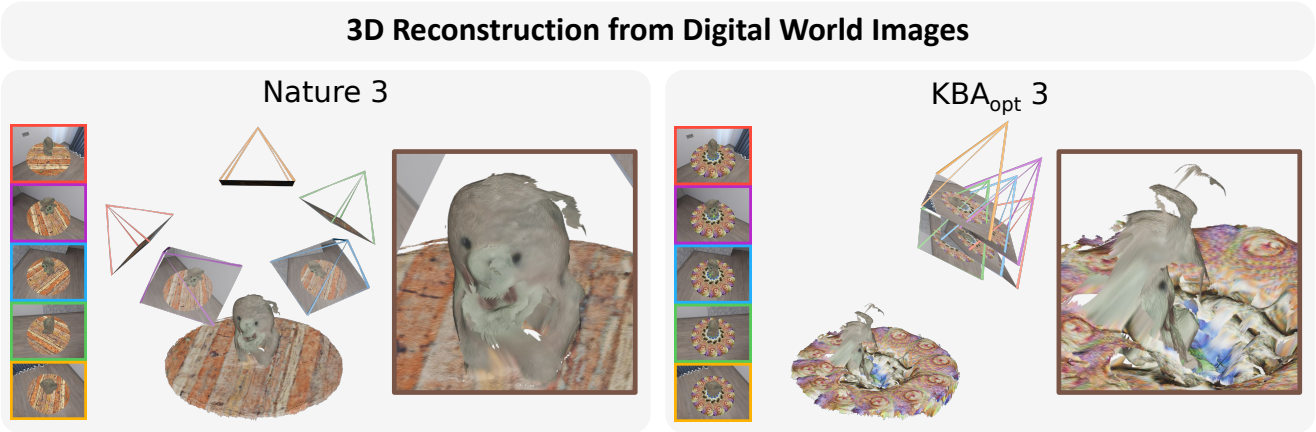


Figure 12. Visualization of 3D reconstruction and pose estimation results on the DUSt3R model for digital-world images of different objects and environments.

M. Additional Visualizations of Pose Estimation Results for KBA_{opt} in the Physical World

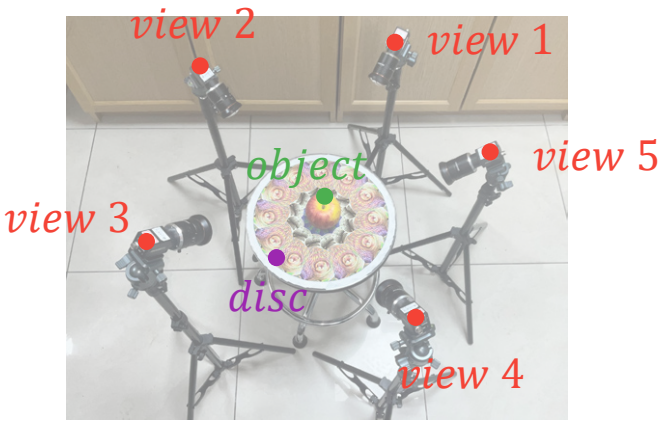


Figure 13. The setup for testing scenes in the physical world.

In this section, we will present additional images of objects captured in the physical world with kaleidoscopic pattern discs, and the corresponding camera pose estimation results on DUSt3R [14], MASt3R [5], RayDiffusion [16], RayRegression [16], PoseDiffusion [13], and RelPose++ [6], as shown in Fig. 14, Fig. 15, and Fig. 16. As shown in Fig. 13, all images were captured by five industrial cameras evenly distributed around the disc, with their lenses directed toward the center at distances ranging from 20 to 50 cm. The experimental results show that, on various objects, our KBA_{opt} method successfully attacks the camera pose estimation results of the DUSt3R model, making the camera pose estimates from different viewpoints almost identical. When transferred to other models with significantly different architectures, our adversarial attack is also able to enforce the majority of the camera pose estimates in the same direction, further demonstrating the effectiveness of our adversarial attack.

Our experiments indicate that, unlike detection and classification tasks, camera pose estimation, as a dense prediction task, **relies more heavily on the independent texture information of various image regions rather than the global context**. This implies that **a small area of the image has limited influence on other areas**, making traditional adversarial attacks using small patches largely ineffective. Therefore, in our physical-world experiments, we ensure that both natural and radially symmetric texture discs occupy a substantial portion of the image. Notably, this fair setup is clearly sufficient to demonstrate the impact of radially symmetric textures on camera pose estimation.

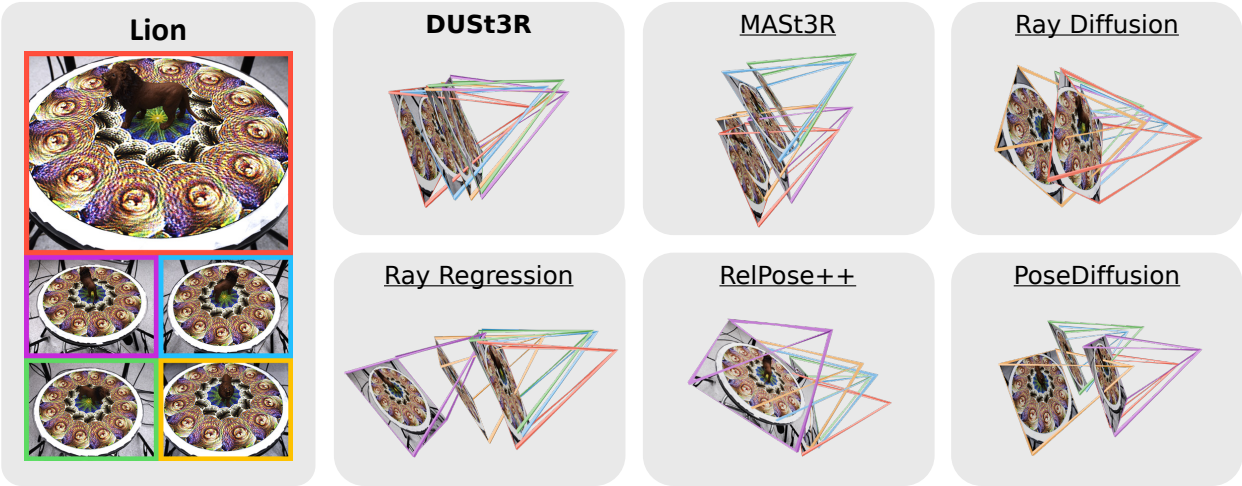


Figure 14. Visualization of five physically captured images, where a toy **lion** is placed above a kaleidoscopic background disc, along with the corresponding camera pose estimation results for various models. The kaleidoscopic background disc, corresponding to the KBA_{opt} method in the main text, is obtained by optimizing the segment image with projected pose consistency loss on DUSt3R.

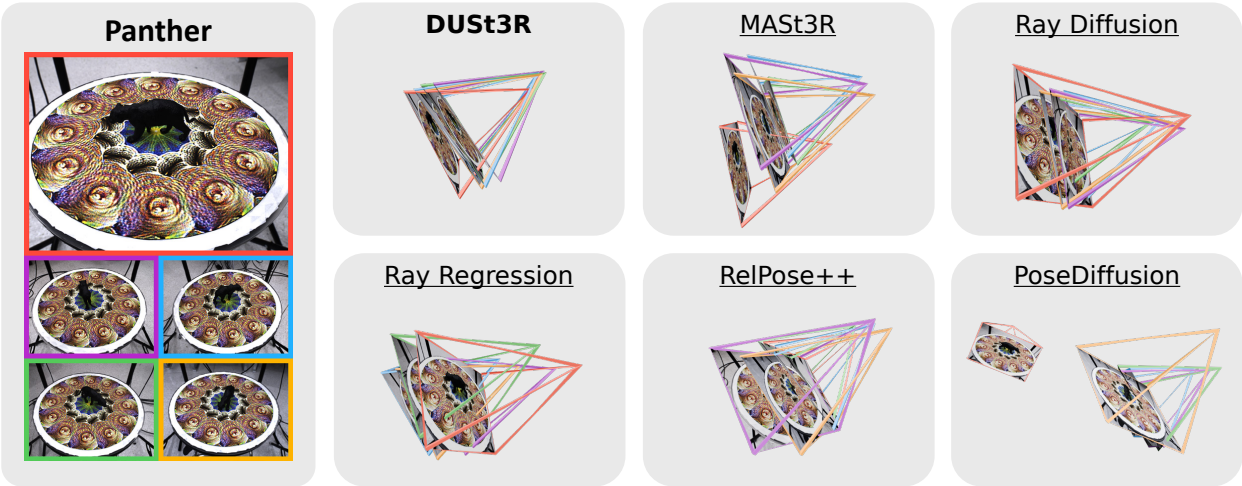


Figure 15. Visualization of five physically captured images, where a toy **panther** is placed above a kaleidoscopic background disc, along with the corresponding camera pose estimation results for various models. The kaleidoscopic background disc, corresponding to the KBA_{opt} method in the main text, is obtained by optimizing the segment image with projected pose consistency loss on DUSt3R.

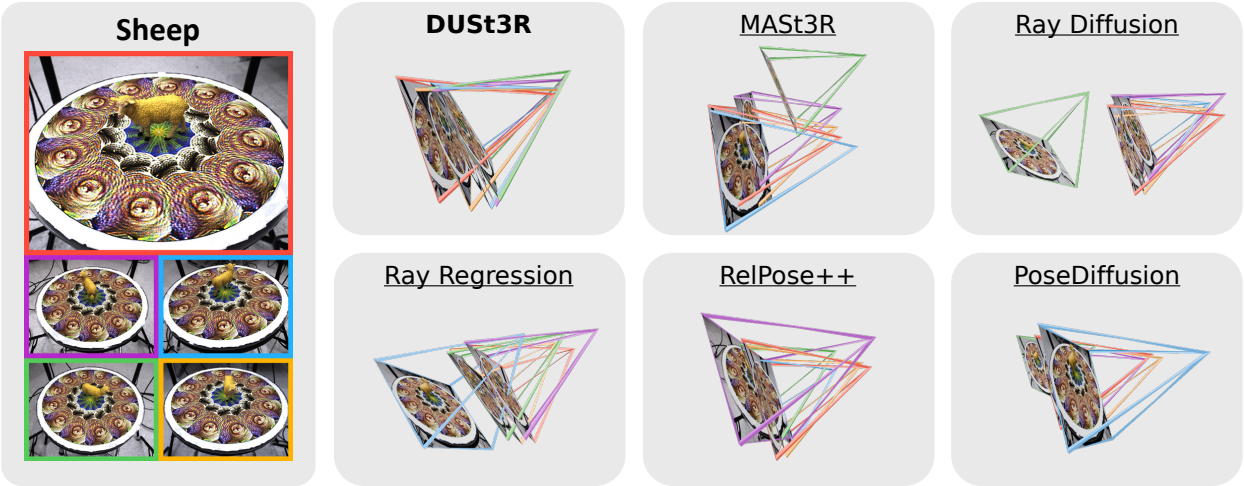


Figure 16. Visualization of five physically captured images, where a toy **sheep** is placed above a kaleidoscopic background disc, along with the corresponding camera pose estimation results for various models. The kaleidoscopic background disc, corresponding to the KBA_{opt} method in the main text, is obtained by optimizing the segment image with projected pose consistency loss on DUSt3R.

References

[1] Adobe Inc. Adobe photoshop, 2024. Accessed: 2024-11-18. 10

[2] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 224–236, 2018. 7

[3] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1625–1634, 2018. 10

[4] Poly Haven. Poly haven: The public 3d asset library. https://polyhaven.com/, 2024. 5, 11

[5] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2024. 5, 15

[6] Amy Lin, Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose++: Recovering 6d poses from sparse-view observations. In *International Conference on 3D Vision (3DV)*, pages 106–115, 2024. 5, 15

[7] Marti Maria. Little color management system, 2024. Accessed: 2024-11-18. 10

[8] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 10901–10911, 2021. 8

[9] John Roberts, Jan A. N. van Aardt, and Fethi B. Ahmed. Assessment of image fusion procedures using entropy, image quality, and multispectral classification. *Journal of Applied Remote Sensing*, 2, 2008. 13

[10] Murray B Sachs, Jacob Nachmias, and John G Robson. Spatial-frequency channels in human vision. *JOSA*, 61(9):1176–1186, 1971. 13

[11] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4938–4947, 2020. 7

[12] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7

[13] Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 9773–9783, 2023. 5, 15

[14] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20697–20709, 2024. 5, 15

[15] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5, 11

[16] Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2024. 5, 15