

The Making and Breaking of Camouflage

Hala Lamdouar¹

Weidi Xie^{1,2}

Andrew Zisserman¹

¹Visual Geometry Group, University of Oxford

²CMIC, Shanghai Jiao Tong University

{lamdouar, weidi, az}@robots.ox.ac.uk

Abstract

Not all camouflages are equally effective, as even a partially visible contour or a slight color difference can make the animal stand out and break its camouflage. In this paper, we address the question of what makes a camouflage successful, by proposing three scores for automatically assessing its effectiveness. In particular, we show that camouflage can be measured by the similarity between background and foreground features and boundary visibility. We use these camouflage scores to assess and compare all available camouflage datasets. We also incorporate the proposed camouflage score into a generative model as an auxiliary loss and show that effective camouflage images or videos can be synthesised in a scalable manner. The generated synthetic dataset is used to train a transformer-based model for segmenting camouflaged animals in videos. Experimentally, we demonstrate state-of-the-art camouflage breaking performance on the public MoCA-Mask benchmark.

1. Introduction

Camouflage has long been a subject of interest and fascination for the scientific community, especially evolutionary biologists who consider it as an excellent example of species adaptation. In order to confuse predators or to hide from prey, and increase their chances of survival in their natural habitat, animals have developed numerous camouflage mechanisms, *e.g.*, disruptive coloration and background matching. Some species have even evolved to develop an adaptive camouflage, *e.g.*, an arctic fox loses its white fur to better match the brown grey of the new season’s landscape. Perhaps the most dramatic camouflage adaptation is the cuttlefish; it changes its patterns dynamically and rapidly as it moves from one spot to the other, constantly adapting and improving its camouflage.

This search for optimal camouflage has inspired numerous works in the computer vision community, such as [12, 27], that tackled camouflage as a problem of optimal texture synthesis, making 3D objects non-detectable in a given scene. Others have addressed camouflage as a highly

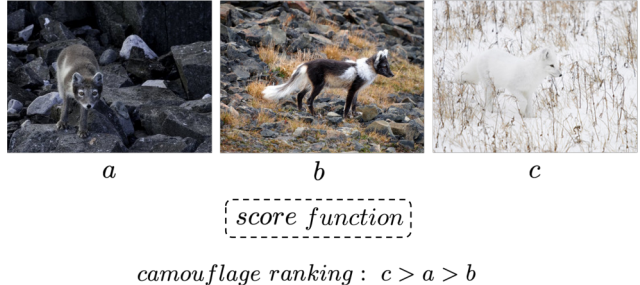


Figure 1: The three images depict the same animal, an Arctic fox, as it adapts its appearance to better blend with the changing landscape of the new season. While images *a* and *c* exhibit better background similarity than image *b*, the fox boundary is more visible in image *a* than image *c*. We assess the effectiveness of camouflage by measuring the degree of ambiguity it creates with respect to its background.

challenging object segmentation task in [9, 15, 20, 23]. Efforts have been made in collecting large scale camouflage datasets [9, 20] with costly annotation. In fact, camouflaged animals often exhibit complex shapes and thin structures that add to the boundary ambiguity and make the annotation highly time-consuming. Fan *et al.* report up to 60min per image to provide accurate pixel-level annotations for their dataset COD10K [9]. While another line of research turned to camouflage breaking in sequences by taking advantage of motion cues [1, 5, 18, 19, 40], the camouflage data scarcity is even more extreme for videos. Recently, Sim2Real [18, 39] training has shown to be very effective for motion segmentation. By training on the optical flows of synthetic videos, these models can generalise to real videos without suffering from the domain gaps.

In this paper, we start by asking the question: “**what are the properties of a successful camouflage ?**” To answer this question, we investigate three scoring functions for quantifying the effectiveness of camouflage, namely, reconstruction fidelity score (S_{R_f}), boundary score (S_b) and intra-image Fréchet score ($d_{\mathcal{F}}$). These scores are later adopted for two critical roles, (i) they are used to assess the relevance of existing camouflage datasets and act as quality-indicator in collecting new camouflage data; (ii)

they can be used as a proxy loss for image and video inpainting/generation, where we establish a synthetic camouflage data generation pipeline with a Generative Adversarial Network (GAN), that can simultaneously generate high-quality camouflage examples and masks of the camouflaged animals. We further train a Transformer-based architecture on the generated camouflage video sequences, and demonstrate a performance boost over training on only the (small scale) real data.

In summary, we make the following contributions: *First*, we introduce three scoring functions to measure the effectiveness of a given camouflage in still images and videos. We use these camouflage scores to rank the images in existing datasets in terms of camouflage success, and also show that the rankings are largely in accordance with human-produced rankings. *Second*, we incorporate the camouflage score into a generative model, establishing a scalable pipeline for generating high-quality camouflage images or videos, along with the pixelwise segmentation mask of the camouflaged animals; *Third*, we show that a Transformer-based model trained on the synthetically generated data can achieve state-of-the-art performance on the MoCA-Mask video camouflaged animal segmentation benchmark.

2. Related work

Camouflage Evaluation. Although there are no previous computational works that directly assess camouflage, as far as we know, there have been a number of human studies. Previous works proposed methods to evaluate camouflage by analysing the human viewers’ eye movements [22, 37] or conducting subjective perceptual experiments [12, 27, 32]. In [12, 27], participants were asked to indicate the camouflaged object and their answers were analysed in terms of accuracy and time needed per image. Similarly, Skurowski *et al.* [32] asked human volunteers to rate CHAMELEON [32] images from 1 to 5 in terms of camouflage effectiveness. They produced a score after compensating for personal bias. We validate our proposed camouflage scoring functions by comparing our rankings to these human-based rankings.

Motion Segmentation. The goal of this task is to partition the frames of a video sequence into background and independently moving objects. Early approaches focused on clustering motion patterns by grouping pixels with similar trajectories [3, 17, 21, 24, 25, 26, 31]. Another line of research tackled the problem by compensating for the background motion [1, 2, 19], via registering consecutive frames, or explicitly estimating camera motion. More recently, in [40], a Transformer-like architecture is used to reconstruct the input flows, and the segmentation masks are generated as a side product, while [6] decomposes the flow field into regions by fitting affine transformations. The most

closely related to ours is [18, 39], where the authors adopt a Sim2Real strategy by training the model on optical flow computed from synthetic videos. These models can generalise towards real videos without fine-tuning. In this paper, we train a model on synthetic camouflage *RGB* sequences.

Fréchet Inception Distance and its variants. Also known as Wasserstein-2 distance [38], Fréchet distance (FD) is a metric quantifying the difference between two probability distributions [10]. Lately, [13] introduced the Fréchet Inception Distance (FID) in the context of generative models. Under the assumption that real images and generated images follow Gaussian distributions, they compute the FD of the two distributions. Note that here, each image is represented by a vector obtained from the last pooling layer of InceptionV3 [35]. Following their steps, [30] adapted FID to the image re-targeting task and introduced the Single Image Fréchet Inception Distance (SIFID). Instead of comparing entire image datasets, they compare two images and consider the distributions of their feature vectors obtained from earlier layer of InceptionV3 [35]. Recently, [7] included FID in the training objective and introduced Fréchet-GAN. Our work takes inspiration from [7, 30] and investigates a FID of regions within the same image as an auxiliary loss.

3. Measuring the effectiveness of camouflage

Assuming there exists a dataset, $\mathcal{D} = \{(I_1, m_1), \dots, (I_N, m_N)\}$, where $I_i \in \mathbb{R}^{H \times W \times 3}$ refers to the images, and $m_i \in \{0, 1\}^{H \times W \times 1}$ denotes a binary segmentation mask of the camouflaged animal. Here, we aim to design a scoring function that takes the image and segmentation mask as input, and outputs a scalar value S that can quantify the effectiveness of camouflage, *i.e.*, how successfully an animal blends into its background, termed as the camouflage score,

$$S : (I_i, m_i) \mapsto s_i \in [0, 1]$$

specifically, for $i \neq j$ and $s_i < s_j$, the animal in I_j , indicated by the mask m_j , is more concealed than the animal in I_i , indicated by the mask m_i . Having such a scoring function enables various applications, *first*, it enables to rank the images from one dataset, in terms of the effectiveness of camouflage, and generate camouflage-relevant statistics for the dataset; *second*, it can serve as the objective for low-level image generation tasks, for example, image inpainting, editing, etc.

In the following sections, we investigate three such scoring functions, specifically, we exploit the animal’s mask and quantify the key aspects that determine the camouflage success both perceptually (Sec. 3.1) and probabilistically (Sec. 3.2). Note that, we take into account the local aspect of camouflage, *i.e.*, instead of processing the entire background region, we only focus on the immediate

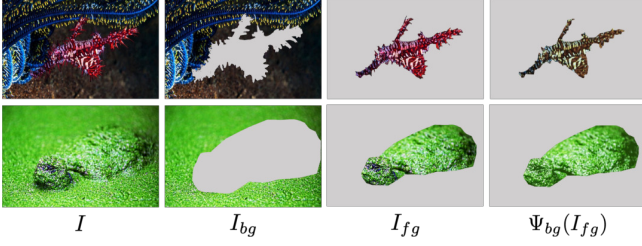


Figure 2: The reconstruction fidelity score S_{R_f} evaluates the similarities between the original foreground I_{fg} and its reconstruction from background features $\Psi_{bg}(I_{fg})$. The top example shows a case where the animal exhibits different visual patterns (color) from its background ($S_{R_f}=0.11$), while the bottom example shows a better background matching ($S_{R_f}=0.82$).

surround of the animal. Hence, when referring to background, we mean the local background and only consider the cropped images centred around the object.

3.1. Perceptual camouflage score

In this section, we define the camouflage score through a perceptual lens, specifically, we measure the foreground and background similarity (Sect. 3.1.1), along with the boundary visibility (Sect. 3.1.2).

3.1.1 Reconstruction fidelity score

We attempt to reconstruct the foreground region with patches from the background, and propose a score to quantitatively measure the discrepancy between the original image and its reconstruction. Intuitively, for successful camouflage, we should be able to reconstruct the foreground perfectly by copying patches from its close neighborhood.

Formally, for a given image (I) and segmentation mask (m), subscript has been ignored for simplicity, we consider a trimap separation and define the foreground and background regions using morphological erosion and dilation of the mask,

$$I_{fg}, I_{bg} = m_{fg} \odot I, m_{bg} \odot I \quad (1)$$

where $m_{fg} = \text{erode}(m)$ and $m_{bg} = 1 - \text{dilate}(m)$.

To reconstruct the foreground region, we replace the foreground patch with the closest one in the background. Here a patch is a $n \times n$ pixel region ($n = 7$ in our case with an overlap of 3) and the patchwise similarity is computed by exploiting the low-level perceptual similarity, *i.e.*, comparing corresponding RGB values. This reconstruction method is inspired by the texture generation and inpainting approaches, such as [8]. In practice, we implement it with fast approximate nearest neighbor search for efficiency.

The reconstruction fidelity score is computed by assessing the difference value between the foreground region and

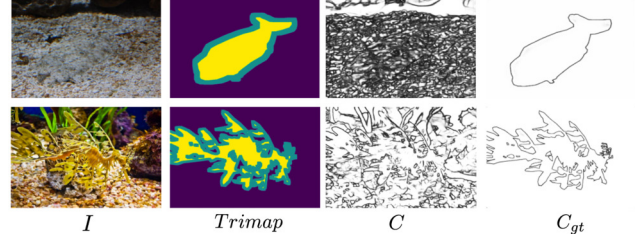


Figure 3: Trimap regions and animal contours. From left to right: Input image; Trimap partition (foreground region in yellow, background in purple, and boundary region in green); Contours C computed on the original image; and ground truth Contours C_{gt} . The top example shows less contour agreement along the boundary region ($S_b=0.72$) than the bottom example ($S_b=0.40$).

its reconstruction, specifically, we count the number of foreground pixels that have been successfully reconstructed from the background:

$$S_{R_f}(I, m) = \frac{1}{N_{fg}} \sum_{(i,j) \in I_{fg}} R_f(i, j) \quad (2)$$

$$R_f(i, j) = \begin{cases} 1 & \text{if } \|I_{fg} - \Psi_{I_{bg}}(I_{fg})\|_2 < \lambda \|I_{fg}\|_2 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$\Psi_{I_{bg}}(\cdot)$ denotes the reconstruction operation, $N_{fg} = |m_{fg}|$ is the total number of pixels in the foreground region, and λ is a threshold ($\lambda=0.2$ in our case). A higher S_{R_f} score, means that the animal’s visual attributes are well represented in the background. Conversely, a low S_{R_f} indicates that the animal’s appearance is composed of unique features that makes it stand out from its surrounding. In Fig. 2, we present examples of camouflage evaluation by reconstruction fidelity.

3.1.2 Boundary visibility score

With the trimap representation, we can easily extract the boundary region, $I_b = m_b \odot I$, where $m_b = (1 - m_{bg}) - m_{fg}$. Here, we aim to measure the animal’s boundary properties, *e.g.*, contour visibility, as they can also provide visually evident cues for camouflage breaking. In particular, we adopt an off-the-shelf contour extraction method [29], and run it on the original image and foreground mask, to generate the images’ contour (C) and ground truth animal’s contour (C_{gt}), as shown in Fig. 3.

We express the agreement between predicted contours C and ground truth contours C_{gt} over the boundary region with the F1 metric:

$$S_b(I, m) = 1 - \text{F1}(m_b \odot C_{gt}, m_b \odot C) \quad (4)$$

This score penalises the boundary pixels that are predicted as contour in both C and C_{gt} . While we do not expect a

perfect contour agreement for visible edges, we consider S_b as a reasonable approximation, given the shape and size of the boundary region as a thin margin centred around the animal. If a boundary pixel is predicted as a contour in C_{gt} but not in C , this means that the animal’s edge is not visible in the original image. If a pixel is predicted as a contour in C but not C_{gt} , this indicates the presence of distracting elements, such as complex texture patterns on the foreground animal, which also affects the visibility of the animal’s contour and therefore improves its camouflage. A perfectly camouflaged animal would have little contour agreement and $S_b = 1$.

To get the final perceptual camouflage score, we linearly combine both reconstruction fidelity score and boundary visibility score:

$$S_\alpha = (1 - \alpha)S_{R_f} + \alpha S_b \quad (5)$$

We will describe our method for setting the weighting parameter α in the Experiment section (Sec. 6).

3.2. Probabilistic scoring function

In addition to using the low-level RGB representation, in this section, we consider the pixelwise representation in the feature space, and propose a differentiable metric that compares the probabilistic distribution between the foreground and background regions, which acts as a proxy for the score S_α described in previous section, and can be used directly as a differentiable loss function in image generation.

We consider the **Intra-Image Fréchet Inception Distance**, specifically, we compute the feature maps for the foreground and background image regions:

$$f_{fg}, f_{bg} = \Phi_v(I_{fg}), \Phi_v(I_{bg}) \quad (6)$$

Here $\Phi_v(\cdot)$ denotes a pre-trained Inception network [34, 35], truncated at early layers. We take inspiration from [30], and adapt the single image Fréchet distance to measure the deviation between feature distributions of different regions within the same image. We adopt the Fréchet hypothesis with respect to our regions, *i.e.*, the features of the foreground and background follow multivariate Gaussian distributions: $f_{fg} \sim \mathcal{N}(\mu_{fg}, \Sigma_{fg})$ and $f_{bg} \sim \mathcal{N}(\mu_{bg}, \Sigma_{bg})$. The intra-image Fréchet distance can be formulated as follows:

$$d_{\mathcal{F}}^2(I, m) = \|\mu_{fg} - \mu_{bg}\|_2^2 + \text{Tr}(\Sigma_{fg} + \Sigma_{bg} - 2(\Sigma_{fg}\Sigma_{bg})^{1/2})$$

Note that, when (I, m) is the output from a generative model, $d_{\mathcal{F}}^2$ is differentiable with respect to its parameters, and can therefore be used as an auxiliary loss for optimising the image generation procedure, *i.e.*, generate effective camouflage examples.

4. Generating camouflage video sequences

In this section, we propose a scalable pipeline for generating images with concealed animals. In particular, we incorporate the differentiable intra-image Fréchet distance in image generation, explicitly optimising the camouflage score of the image. We first detail the static image generation framework, then describe how we use the images it generates to create camouflage video sequences.

4.1. Camouflage image generation

In order to generate the image with a camouflaged animal, and its corresponding segmentation mask, we adopt a Generative Adversarial Network (GAN), where a generator is fed with a latent vector $z \sim \mathcal{N}(0, 1)$ and learns to predict the pair of realistic image and segmentation mask, such that a discriminator cannot differentiate it from real samples.

Let (x_i, m_i) denote an image and segmentation pair sampled from the training dataset, a subset of COD10K [9] is used here. We adopt the conventional label notation, *i.e.* $y = 1$ for (real) examples sampled from the training dataset and $y = 0$ for (fake) examples from the generator, and consider the generative adversarial loss functions [11]:

$$\begin{aligned} \mathcal{L}_D &= E_{(x_i, m_i)}[-\log(D(x_i, m_i))] + E_z[-\log(1 - D(G(z)))] \\ \mathcal{L}_G &= E_z[-\log(D(G(z)))] \end{aligned}$$

To enforce coherence between images and their segmentation masks, we feed the discriminator with additional fake pairs, consisting of real images coupled with unpaired real masks. This can be formulated as an additional coherence loss term that minimises the probability of assigning incorrect labels, *i.e.* $y = 1$, to (x_i, m_j) sampled from the training set such that $i \neq j$:

$$\mathcal{L}_{\text{Coh}} = E_{(x_i, m_j)_{i \neq j}}[-\log(1 - D(x_i, m_j))] \quad (7)$$

To increase the camouflage effectiveness in the generated examples, we adopt the intra-image Fréchet distance in the generator loss, as an auxiliary loss term, and train our camouflage image generator with the following loss:

$$\mathcal{L}_{\tilde{D}} = \mathcal{L}_D + \mathcal{L}_{\text{Coh}} \quad \mathcal{L}_{\tilde{G}} = \mathcal{L}_G + \beta d_{\mathcal{F}}^2 \quad (8)$$

We present an overview of our data generation framework in Fig. 4.

4.2. Camouflage video generation

Given a camouflage image and corresponding segmentation mask, generated using the method above, we can create video sequences by applying different motions to the foreground and background, in a similar manner to [18]. Specifically, we first inpaint the backgrounds with an off-the-shelf model proposed by Suvorov *et al.* [33] and overlay the synthetic animal (extracted from the original generated image

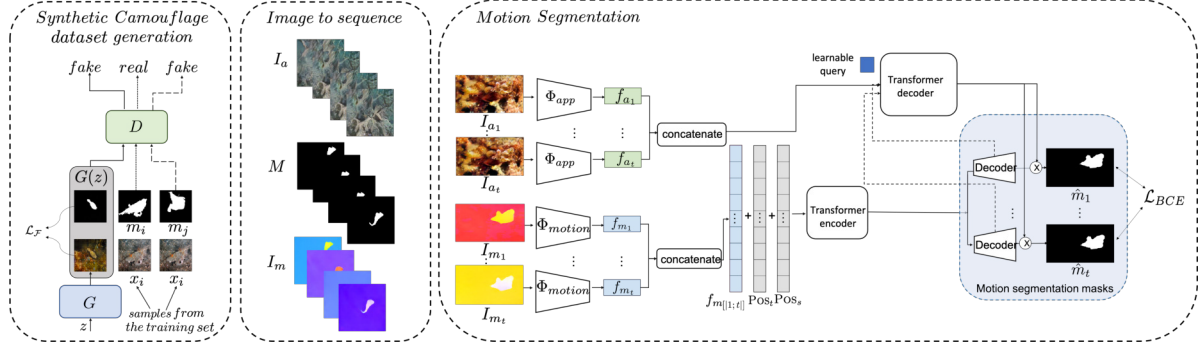


Figure 4: Our framework consists of a synthetic camouflage data generation pipeline (left), where we train a generator G , in a GAN setting, to create camouflage images and masks, while encouraging camouflage effectiveness via minimising $\mathcal{L}_{\mathcal{F}}$. The generated samples are then transformed into synthetic video sequences (middle), following our method presented in Sec. 4.2. The transformer-based motion segmentation architecture (right), for camouflage breaking in videos, is trained on the synthetic video sequences. The architecture is described in Sec. 5.

by using the mask) at a random location within the image in the first frame, then following a random translational trajectory in the following frames. We incorporate a different translational motion for the background and include random sub-sequences where the foreground and background undergo the same motion to simulate momentarily static objects. As we train our generator with intra-image Fréchet loss ($\mathcal{L}_{\mathcal{F}}$), the generated images exhibit strong similarities between the foreground and the background, and even if the object is placed at a different location from the original, we expect it to remain concealed within its surrounding.

5. Learning to break camouflage

In this section, we train a transformer-based architecture on the synthetic dataset, and demonstrate its effectiveness for breaking the camouflage in real videos, for example, MoCA [19]. We build on two previous architectures, namely, the motion segmentation from [18] and the Search Identification Network from [9]. While the first architecture processes sequences of optical flow, the second takes single RGB images as inputs and treats them separately. Our proposed model, shown in Fig. 4, takes both the optical flow and RGB frame sequences as input. The flow is computed with RAFT [36] from the RGB sequence, and processed with a motion encoder, followed by a transformer encoder. The RGB sequence is processed with the appearance encoder from [9], pre-trained for framewise camouflage segmentation. Then both streams are aggregated as inputs to a Mask2Former-like [4] transformer decoder and pixel decoder to produce the high resolution segmentation mask from the motion stream.

Motion encoder. To encode the motion cues, we use a light-weight convNet that takes as input a sequence of optical flows, $\mathbf{I}_m = \{I_{m_1}, I_{m_2}, \dots, I_{m_t}\} \in \mathcal{R}^{t \times c_0 \times h \times w}$ and out-

puts motion features:

$$\{f_{m_1}, f_{m_2}, \dots, f_{m_t}\} = \Phi_{motion}(\mathbf{I}_m)$$

where each flow frame is separately embedded.

Motion transformer encoder. A transformer encoder takes as input the motion features, concatenated along the sequence, together with learned spatial and temporal positional encodings ($\text{Pos}_s, \text{Pos}_t$), as indicated in Fig 4. Pos_t specifies the frame, and Pos_s specifies the position in the spatial feature map output of the motion encoder. The output of the transformer is a set of enriched motion features.

Appearance encoder. Here, we adopt a SInet-v2 [9] architecture, that encodes the RGB sequence, $\mathbf{I}_a = \{I_{a_1}, I_{a_2}, \dots, I_{a_t}\} \in \mathcal{R}^{t \times c_1 \times h \times w}$ into appearance features:

$$\{f_{a_1}, f_{a_2}, \dots, f_{a_t}\} = \Phi_{app}(\mathbf{I}_a)$$

Again, each RGB frame is processed separately by SInet.

Transformer decoder. A transformer-based decoder takes the output of the motion transformer encoder together with the appearance features and a learnable query for the mask embedding. In a similar manner to Mask2Former [4], the query attends to multiple resolutions of the motion features concatenated with the appearance features and produces a mask embedding for the moving object.

Pixel Decoder. Similarly to the pixel decoder in Mask2Former, a light-weight convNet decoder is used with skip-connections to recover high-resolution segmentation masks $\{\hat{m}_1, \hat{m}_2, \dots, \hat{m}_t\}$ from the motion features and the mask embedding. This is shown as the blue box in Fig. 4.

Training objective. We train the motion segmentation model on our synthetic video sequences using the binary cross-entropy loss $\mathcal{L}_{BCE}$.

6. Experiments

In this section, we start by introducing the datasets involved in this paper, followed by the implementation details. Our experiments present a thorough analysis of the proposed camouflage scores and demonstrate their effectiveness in our training framework.

6.1. Datasets

Here we describe the publicly available camouflage datasets that we included in our experiments, as shown in Fig. 5, as well as the synthetic camouflage datasets that we generated.

CHAMELEON [32] is one of the first camouflage image datasets. It contains 76 images collected using Google image search with the ‘camouflaged animal’ keyword and include ground truth manual annotations.

CAMO [20]. The Camouflaged Object dataset consists of 2500 images collected from the internet, of which 1250 images (1000 training sub-set and 250 testing sub-set) contain at least one camouflage instance from 8 categories with manual pixelwise annotations provided.

COD10K [9]. COD10K contains 10,000 images collected from photography websites of which 5066 depict camouflaged animals (3040 training sub-set and 2026 testing sub-set), organised into 10 classes, and 78 sub-classes (69 camouflaged). Note that, in our camouflage evaluation experiments we only used the camouflage subset of this dataset.

Camouflaged Animals [1]. This dataset consists of 9 video sequences of camouflage animals from 6 categories.

MoCA [19] is the first large-scale camouflage video dataset. It contains 141 video sequences (37K frames) totalling 67 animal categories. Recently, other works have curated this dataset by selecting only the videos with more prominent motion (locomotion) [40], and others have provided dense pixel annotation in MoCA-Mask [5]. We use the latter version in our experiments.

Camouflaged cuboids [12, 27]. This dataset was created for texture synthesis and consists of multiple-view scenes, where cuboids were placed at a predefined location then synthetically camouflaged. In our evaluation, we consider the 4-views generated textures [12] from 36 scenes as well as the cuboids masks.

Synthetic Camouflage Images. Using the method described in Sec. 4.1, we generate a synthetic camouflage dataset, of 5K frames, by discriminating against real camouflage image and annotation masks from COD10K.

Synthetic Camouflage Videos. We generate 1K sequences of 30 frames each, incorporating static sequences using the framework described in Sec. 4.2. We split these into 800 sequences for training and 200 for testing.



Figure 5: **Randomly sampled examples from the camouflage datasets included in our work.** For the video datasets, Camouflaged Animals and MoCA-Mask, we show an example sequence. For the multiple-view dataset Camouflaged cuboids (bottom), we show example views from a scene with the 4-view texture synthesis method from [12].

6.2. Implementation details

In our experiments, for a given camouflage example, the kernels for the morphological operations are selected from a range of values [1,10], so that the resulting annotation mask

is reduced by 20% for erosion and extended by 20% for dilation. This allows the erosion/dilation to be adapted to the size of the camouflaged animal, while always keeping a reasonably large region for addressing the boundary effects.

For our synthetic camouflage image generation, we adopt a Style-GAN architecture [41] and train it on the camouflage images from COD10K to generate 256×256 images and masks pairs. We use $\beta = 0.1$ to weight our intra-image Fréchet auxiliary loss. For numerical stability, we adopt the Newton Schultz iterations when calculating the matrix square root term in Intra-Image Fréchet Inception Distance, with $T = 100$ iterations.

When generating the image sequences, we compute optical flows with RAFT [36], and train our moving camouflaged animal segmentation model. We use batches of size 2 and sequences of size 5. We first train on our synthetically generated dataset with a learning rate of 5×10^{-4} for 500 iterations then fine-tune on the training subset of MoCA-Mask.

6.3. Results

This section presents qualitative and quantitative results, to demonstrate the effectiveness of our score functions and the benefit of including $d_{\mathcal{F}}^2$ in the training loop.

6.3.1 Evaluation on camouflage effectiveness

Ranking camouflage images in COD10K. We can rank the images based on the effectiveness of the animal’s camouflage, with our proposed scoring functions. In the left part of Fig. 6, we show the four best scored images from the large scale COD10K dataset. We can make the following observations: (i) the best examples with respect to S_{R_f} (top), which favours the **background matching** of the animals without taking into account the boundary region, all exhibit visible boundaries, with the exception of image (a). This is especially noticeable in the rabbit example (d) along the ears and shadow regions; (ii) the best four examples with respect to the **boundary score** S_b are all from the caterpillar subclass, mostly the baron type. These insects have thin, and transparent boundaries, that makes them the perfect candidates for this score. However, with the exception of (a) and (c), the animals still stands out as they exhibit colors and patterns that are not present within their background; (iii) the S_α **combining both scores** shows highly effective camouflage examples. In contrast, the right part of Fig. 6 shows the lowest scored images for each approach. For S_{R_f} , we find examples with low background matching in the first row, with two ant examples, *i.e.*, (e), (f). While the boundary score penalises examples with high contrasts between the animal and its background, that results in more visible contours. This is also the case for the lowest S_α score examples.

Dataset level camouflage comparison. We compute the

camouflage score for all the camouflage image and video datasets and report the results in Tab. 1. For the image datasets, the dataset level score is computed as the mean of the image scores; for the video dataset the score is first computed per frame, and per-video average computed, then the dataset level score is the mean of the averages. Our experiments show that MoCA-Mask [5, 19] contains the most successful camouflages according to our scoring. While COD10K [9] subsets are balanced in terms of camouflage effectiveness, we find that, for CAMO [20], the test subset yields higher scores than the training subset and therefore contains more challenging examples.

We note that Camouflaged cuboids dataset yields higher S_{R_f} and S_α scores compared to our synthetic datasets. This is due to the fact that the model used in [12, 27] is only trained to produce optimal texture for a predefined region (cuboid) in a particular location of a given scene. However, our model learns to output a *new* image with a randomly located camouflaged object of random shape, thus offering more scalability and diversity compared to the cuboids dataset which cannot be used to train a model for breaking real camouflage.

Comparison to human-produced rankings. We compare our rankings to the human scoring system based on the rating of CHAMELEON [32] and the time-to-find for Camouflaged cuboids [12]. To search for the optimal α parameter in Equation (5), we select 15 images from CHAMELEON and compare their S_α ranking with ground truth using kendall- τ metric [16]. For $\alpha = 0.35$, we obtain kendall- $\tau = 0.51$ on this validation set that we excluded from our test reported in Tab. 2. We can draw the following observations: (i) we find that the boundary score S_b produces more agreement with the ground truth ranking than S_{R_f} , suggesting that human observers tend to pay more attention to contour visibility than background matching; (ii) while comparing to the ranking from Camouflaged cuboids, we found negative correlations for S_{R_f} and $d_{\mathcal{F}}$, we conjecture that this may be due to the nature of the dataset in [12], *i.e.*, all the textured cuboids are synthetically generated with very high background matching; (iii) for both experiments, we obtain the strongest agreement with the S_α score combining both background similarity and boundary blending.

Further analysis of the synthetic datasets. We adopt the FID [13] and IS [30] metrics in Table 4. These metrics assess: (i) the similarity between the set of synthetic images output by the generator and the set of real images used to train it (FID); and (ii) the clarity of the object and the diversity of images in terms of classes (IS). While they are not intended for measuring camouflage success, which is the focus of this work, IS could somewhat be (inversely) linked to S_b as the object clarity and boundary visibility are close. Note that the object clarity (IS) is decreased with $\mathcal{L}_{\mathcal{F}}$ which is the intended purpose of improving camouflage, however,

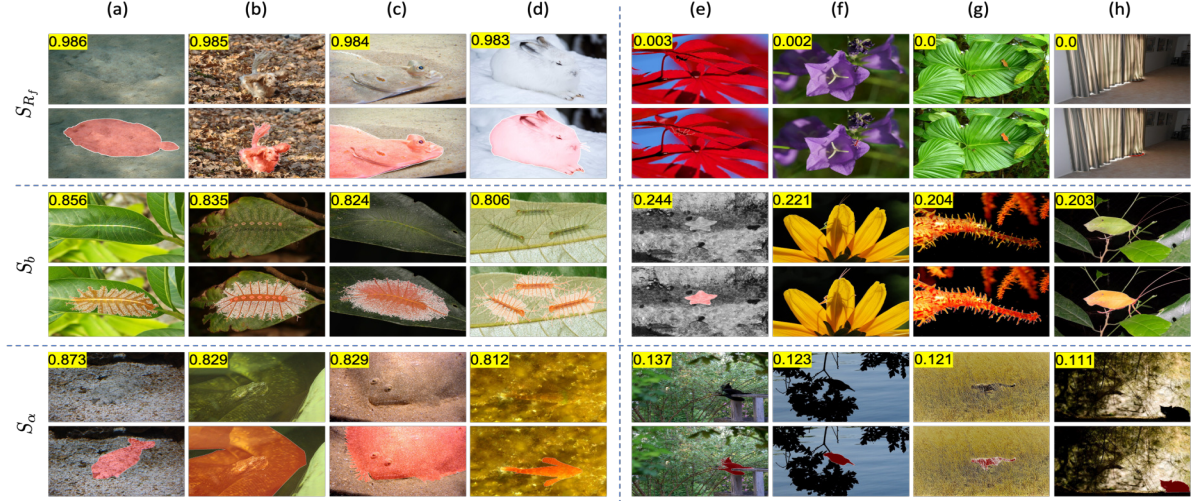


Figure 6: **Left:** Top-4 scored examples from COD10K for S_{Rf} (top), S_b (middle) and S_α (bottom). **Right:** Lowest scored examples from COD10K. For each example, we show the obtained score and the corresponding ground truth mask.

Datasets	Data type	$S_{Rf} \uparrow$	$S_b \uparrow$	$S_\alpha \uparrow$	$d_{\mathcal{F}}^2 \downarrow$
CHAMELEON [32]	Image	0.694	0.445	0.607	0.70
CAMO Train [20]	Image	0.672	0.451	0.595	1.01
CAMO Test [20]	Image	0.683	0.470	0.608	1.00
COD10K Train [9]	Image	0.655	0.433	0.577	0.90
COD10K Test [9]	Image	0.657	0.431	0.578	0.90
Camouflaged Animals [1]	Video	0.674	0.536	0.626	1.60
MoCA-Mask Train [5, 19]	Video	0.850	0.443	0.707	1.14
MoCA-Mask Test [5, 19]	Video	0.733	0.464	0.639	2.51
Camouflaged cuboids[12, 27]	Multi-view	0.894	0.433	0.733	6.2
Syn. Camouflage w.o. $\mathcal{L}_{\mathcal{F}}$	Image	0.608	0.432	0.546	1.36
Syn. Camouflage w. $\mathcal{L}_{\mathcal{F}}$	Image	0.679	0.447	0.598	1.13
Syn. Camouflage Video	Video	0.658	0.430	0.578	1.18

Table 1: Results of the proposed scores on natural camouflage datasets (top) and synthetically generated camouflage datasets (bottom). We report the mean scores for the single image datasets. For the video and multiple view datasets, we compute the mean per sequence and scene respectively. *Syn. Camouflage w. $\mathcal{L}_{\mathcal{F}}$* refers to the synthetic image dataset that we generated while minimising $\mathcal{L}_{\mathcal{F}}$ and *Syn. Camouflage Video* its sequence version.

this effect is not maintained in the sequence generation as the animal changes location within its background.

dataset	FID↓	IS↑
Syn. Camouflage w.o. $\mathcal{L}_{\mathcal{F}}$	75.01	3.49
Syn. Camouflage w. $\mathcal{L}_{\mathcal{F}}$	76.34	3.13
Syn. Camouflage Video	86.42	3.85

Table 4: **Overall quality of our synthetic datasets.**

6.3.2 Model ablations

On the effectiveness of $\mathcal{L}_{\mathcal{F}}$. Fig. 7 shows the generated samples from the GAN pipeline, trained with and without the intra-image Fréchet Distance. Our generator is able to produce realistic object masks of complex shapes and thin structures similar to those usually encountered in the camouflage datasets. The produced images exhibit rich back-

Scores	CHAMELEON [32]	Camouflaged cuboids[12, 27]
S_{Rf}	0.01	-0.07
S_b	0.03	0.42
S_α	0.42	0.41
$d_{\mathcal{F}}^2$	0.10	-0.17

Table 2: Kendall- τ metric between rankings produced via our scores and human scoring ground truth.

Model	Training dataset	Appearance Encoder	Transformer Encoder	Transformer Aggregation	mIoU↑	$F \uparrow$	$E \uparrow$	MAE↓
A	S				14.5	20.4	57.3	9.4
B	MM				15.3	20.6	57.3	5.1
C	S+MM				16.0	21.8	59.8	4.3
D	S+MM	✓			20.5	22.6	59.8	3.8
E	S+MM	✓	✓		23.0	23.5	57.0	1.6
F	S+MM	✓		✓	22.7	23.2	58.8	3.6
G	MM	✓	✓	✓	23.4	24.7	62.0	2.4
H	S+MM	✓	✓	✓	30.8	34.3	77.0	1.8

Table 3: Ablation study on the different components of the motion segmentation architecture and training datasets. We evaluate on MoCA-Mask for all experiments. (S: our synthetic dataset, MM: MoCA-Mask).

grounds with realistic nature-like elements mimicking rocks and coral structures present in the real camouflage data. Adding the $\mathcal{L}_{\mathcal{F}}$ loss results in generating images with better blending in the background, both qualitatively and quantitatively, as shown by the S_α computed for both datasets generated with and without the intra-image Fréchet Distance in Tab. 1. We refer the reader to the supplementary material for more examples of generated samples.

On the architecture design and gain from the synthetic dataset. We present in Tab. 3 an ablation study detailing the gain from the main components of our architecture, as well as the benefit of first training on our synthetic dataset then fine-tuning on MoCA-Mask (S+MM), for model H, as opposed to only training on MoCA-Mask (MM), for G.

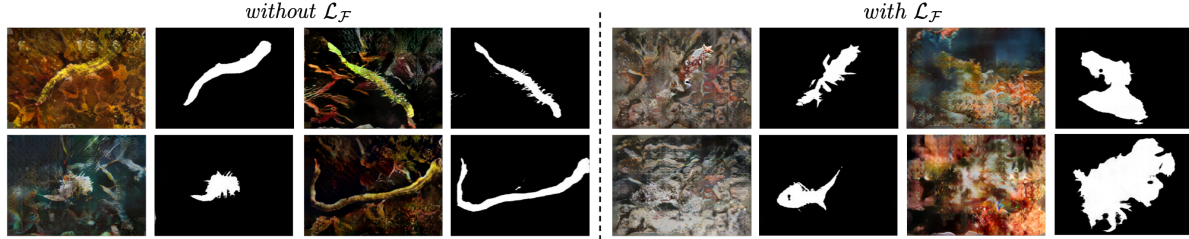


Figure 7: **Generated camouflage images and masks:** For the examples on the right, the generator was trained with $\mathcal{L}_{\mathcal{F}}$.

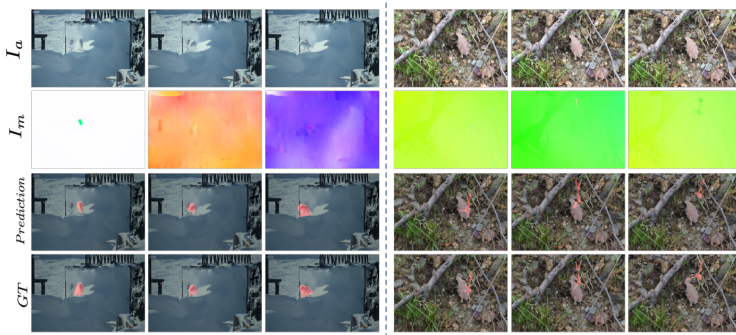


Figure 8: **Qualitative results on MoCA-Mask.** From top to bottom: Appearance sequence I_a , flow sequence I_m , predicted segmentations and ground truth segmentations.

Model	RGB	Motion	mIoU $\uparrow$	$F\uparrow$	$E\uparrow$	MAE $\downarrow$
SINet [9]	✓		20.2	23.1	69.9	2.8
SINet-V2 [9]	✓		18.0	20.4	64.2	3.1
SegMaR [14]	✓		12.2	22.5	80.3	4.5
ZoomNet [28]	✓		18.8	28.7	70.8	2.5
SLTNet [5]	✓	✓	27.2	31.1	75.9	2.7
MG [40]		✓	12.7	16.8	56.1	6.7
Ours-flow		✓	17.8	21.5	60.7	3.7
Ours	✓	✓	30.8	34.3	77.0	1.8

Table 5: **Results on MoCA-Mask.** Ours-flow refers to the optical flow only version of our architecture, excluding the appearance stream.

6.3.3 Results on MoCA

We train our model on the synthetically generated camouflage images and fine-tune on the training set of MoCA-Mask. Tab. 5 presents our camouflage object segmentation results on the test set of MoCA-Mask. Ours refers to our proposed method with SINet-V2 as the appearance encoder and Ours-flow refers to the optical flow only version of our architecture, excluding the appearance stream. Our main model outperforms RGB and Motion-based methods on MoCA-Mask. Fig. 8 presents qualitative results of our segmentation model. Note that our model is robust to degraded optical flow and static animals.

6.4. Limitations

While our image generation method encourages background matching through the $\mathcal{L}_{\mathcal{F}}$ loss term minimization, the sequence generating is not guaranteed to maintain object concealment. A possible solution could be to use the proposed S_{α} camouflage score to curate the generated sequence dataset and filter out the most visible objects.

Our proposed camouflage scores use the ground truth annotation and analyse the different regions it defines. We found that our scoring system penalises specific cases of camouflage by occlusion, where elements from the background are partially occluding the animal and helping improve its camouflage. For instance, the grass in example (g) from the last row of Fig. 6 is not treated as part of the animal and therefore not considered in our background similarity

assessment. One can argue that this is due to the ambiguity in the provided annotations and for such cases, extra amodal annotation should be also considered.

7. Conclusion

We present three score functions for computationally assessing the effectiveness of camouflage in images and videos. By evaluating the similarity with background and the boundary visibility, our combined perceptual score is strongly correlated with human perceptual ranking systems on two different datasets. We demonstrate that training a generative model with our differentiable camouflage function improves the effectiveness of generated camouflage examples and can be used to generate challenging synthetic camouflage datasets to train models to break camouflage.

Acknowledgement

We are grateful to Przemysław Skurowski for providing the human perception study data on the CHAMELEON dataset and Andrew Owens, Rui Guo and Oscar de Lima for providing the data for the Camouflaged cuboids dataset. We thank Tengda Han for fruitful discussions. This research is supported by the UK EPSRC funded CDT in Autonomous Intelligent Machines and Systems (AIMS), the EPSRC Programme Grant VisualAI EP/T028572/1, a Schlumberger Studentship, and a Royal Society Research Professorship. WX is supported by the National Key R&D Program of China (No. 2022ZD0161400).

References

- [1] Pia Bideau and Erik Learned-Miller. It’s moving! a probabilistic model for causal motion segmentation in moving camera videos. In *ECCV*, pages 433–449, 2016.
- [2] Pia Bideau, Rakesh R Menon, and Erik Learned-Miller. Moa-net: self-supervised motion segmentation. In *Proc. ECCV Workshop*, 2018.
- [3] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *European conference on computer vision*, pages 282–295, 2010.
- [4] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, pages 1290–1299, 2022.
- [5] Xuelian Cheng, Huan Xiong, Deng-Ping Fan, Yiran Zhong, Mehrtash Harandi, Tom Drummond, and Zongyuan Ge. Implicit motion handling for video camouflaged object detection. In *CVPR*, pages 13864–13873, 2022.
- [6] Subhabrata Choudhury, Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht. Guess What Moves: Unsupervised Video and Image Segmentation by Anticipating Motion. In *Proc. BMVC*, 2022.
- [7] Khoa D Doan, Saurav Manchanda, Fengjiao Wang, Sathiya Keerthi, Avradeep Bhowmik, and Chandan K Reddy. Image generation via minimizing fréchet distance in discriminator feature space. *arXiv preprint arXiv:2003.11774*, 2020.
- [8] A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *Proc. ICCV*, pages 1039–1046, Sep 1999.
- [9] Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao. Concealed object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [10] Maurice Fréchet. Sur la distance de deux lois de probabilité. *Comptes Rendus Hebdomadaires des Seances de L’Academie des Sciences*, pages 689–692, 1957.
- [11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.
- [12] Rui Guo, Jasmine Collins, Oscar de Lima, and Andrew Owens. Gannouflage: 3d object nondetection with texture fields. In *arXiv preprint arXiv:2201.07202*, 2022.
- [13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.
- [14] Qi Jia, Shuilian Yao, Yu Liu, Xin Fan, Risheng Liu, and Zhongxuan Luo. Segment, magnify and reiterate: Detecting camouflaged objects the hard way. In *CVPR*, pages 4713–4722, 2022.
- [15] Nobukatsu Kajiura, Hong Liu, and Shin’ichi Satoh. Improving camouflaged object detection with the uncertainty of pseudo-edge labels. In *ACM Multimedia Asia*, pages 1–7, 2021.
- [16] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.
- [17] Margret Keuper, Bjoern Andres, and Thomas Brox. Motion trajectory segmentation via minimum cost multicut. In *ICCV*, pages 3271–3279, 2015.
- [18] Hala Lamdouar, Weidi Xie, and Andrew Zisserman. Segmenting invisible moving objects. In *Proc. BMVC*, 2021.
- [19] Hala Lamdouar, Charig Yang, Weidi Xie, and Andrew Zisserman. Betrayed by motion: Camouflaged object discovery via motion segmentation. In *Proc. ACCV*, 2020.
- [20] Trung-Nghia Le, Tam V Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabran network for camouflaged object segmentation. *Computer Vision and Image Understanding*, pages 45–56, 2019.
- [21] José Lezama, Karteek Alahari, Josef Sivic, and Ivan Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *CVPR*, pages 3369–3376, 2011.
- [22] Chiuhsiang Joe Lin, Chi-Chan Chang, and Yung-Hui Lee. Evaluating camouflage design using eye movement data. *Applied ergonomics*, pages 714–723, 2014.
- [23] Haiyang Mei, Ge-Peng Ji, Ziqi Wei, Xin Yang, Xiaopeng Wei, and Deng-Ping Fan. Camouflaged object segmentation with distraction mining. In *CVPR*, pages 8772–8781, 2021.
- [24] Peter Ochs and Thomas Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, 2011.
- [25] Peter Ochs and Thomas Brox. Higher order motion models and spectral clustering. In *CVPR*, pages 614–621, 2012.
- [26] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. volume 36, pages 1187–1200, 2013.
- [27] Andrew Owens, Connelly Barnes, Alex Flint, Hanumant Singh, and William Freeman. Camouflaging an object from many viewpoints. In *CVPR*, pages 2782–2789, 2014.
- [28] Youwei Pang, Xiaoqi Zhao, Tian-Zhu Xiang, Lihe Zhang, and Huchuan Lu. Zoom in and out: A mixed-scale triplet network for camouflaged object detection. In *CVPR*, pages 2160–2170, 2022.
- [29] Xavier Soria Poma, Edgar Riba, and Angel Sappa. Dense extreme inception network: Towards a robust cnn model for edge detection. In *Proc. WACV*, pages 1923–1932, 2020.
- [30] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *ICCV*, pages 4570–4580, 2019.
- [31] Josef Sivic, Frederik Schaffalitzky, and Andrew Zisserman. Object level grouping for video shots. In *Proc. ECCV*. Springer-Verlag, 2004.
- [32] Przemysław Skurowski, Hassan Abdulameer, J Błaszczyk, Tomasz Depta, Adam Kornacki, and P Koziel. Animal camouflage analysis: Chameleon database. *Unpublished manuscript*, 2018. URL: <https://www.polsl.pl/rau6/chameleon-database-animal-camouflage-analysis/>.
- [33] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. *arXiv preprint arXiv:2109.07161*, 2021.
- [34] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, 2015.
- [35] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception ar-

- chitecture for computer vision. In *CVPR*, pages 2818–2826, 2016.
- [36] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.
 - [37] Jolyon Troscianko, John Skelhorn, and Martin Stevens. Quantifying camouflage: how to predict detectability from appearance. *BMC evolutionary biology*, 17(1):1–13, 2017.
 - [38] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.
 - [39] Junyu Xie, Weidi Xie, and Andrew Zisserman. Segmenting moving objects via an object-centric layered representation. In *NeurIPS*, 2022.
 - [40] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *Proc. ICCV*, 2021.
 - [41] Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. Styleswin: Transformer-based gan for high-resolution image generation. In *CVPR*, pages 11304–11314, 2022.