On Segmenting Moving Objects with Minimal Visibility



Hala Lamdouar

Wadham College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Michaelmas 2022

"The landscape, by a strange optical illusion,
seems to disappear entirely in its body.
It is no longer the snow leopard that is camouflaged in the landscape,
but the world that has been incorporated into it."

"Le paysage, par une étrange illusion d'optique,
semble se résorber tout entier dans son corps.
Ce n'est plus la panthère qui est camouflée dans le paysage,
mais le monde qui s'est incorporé à elle."

Sylvain Tesson

Abstract

In this thesis, we develop methods to exploit motion as a key attribute for detecting visually challenging moving objects. In a video observation, two categories of motion can be noted: a background motion, usually resulting from the camera movement, and the motion of the foreground objects present in the scene. We first revisit geometrical approaches to motion segmentation by investigating registration to compensate for camera motion. To this end, we model and learn to predict the background transformation between consecutive frames solely from optical flow input. We demonstrate that this effectively enhances the moving objects in the registered image difference and improves the segmentation performance. To validate our motion segmentation models, we collect a highly challenging dataset for Moving Camouflaged Animals (MoCA), which is today the largest benchmark for video camouflage object detection.

Second, we explore object-centric approaches to motion segmentation. In the first approach, we investigate a dual architecture that learns both the conventional modal segmentation, *i.e.* segmenting the visible part of the object, and the amodal segmentation, *i.e.* segmenting the object as a whole including the occluded part. To train our model, we develop a pipeline for creating large scale synthetic datasets, in the optical flow space, simulating various types of object shapes and motion and incorporating occlusion. We demonstrate that the motion segmentation model, trained from scratch on this synthetic dataset, generalises well to the real world data, such as DAVIS2016, SegTrackV2 and our challenging MoCA, outperforming vision-based and motion-based methods trained on costly annotated real datasets. In the second approach, we learn a layered representation of the optical flow using slot attention in a self-supervised manner.

Finally, we extend our synthetic dataset generation from the optical flow space to the RGB space and train a generative model to artificially create camouflage examples. We further define the factors that contribute to the effectiveness of a camouflage and propose score functions to quantify its challenge. We demonstrate that enforcing camouflage effectiveness while training the generative model results in improving the quality of the generated camouflage dataset, which, when used as a training set, boosts the performance of the motion segmentation model on the challenging MoCA benchmark.

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Hala Lamdouar, December 2022.

Acknowledgement

I am eternally grateful to my advisor Andrew Zisserman for his invaluable guidance, and support, constantly motivating me and inspiring me to be a better researcher. I am deeply grateful to my co-advisor Weidi Xie for his kind support and dedication, especially the past year, from the other side of the world.

I would like to thank my amazing collaborators from the Visual Geometry Group, Charig Yang and Erika Lu, and all the members of the group, whose enthusiasm and dedication are truly inspiring.

I am deeply grateful to the UK EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems (AIMS) and Schlumberger for funding my thesis and especially Wendy Poole for being a truly wonderful administrator and person.

I am also grateful to my generous friends Inès, Ada, Alaa, Chia-Man, Heba, Sarah, Ech-Chaymae, for their warm presence and kind support, especially in the challenging times.

Finally, I am forever indebted to my marvellous family: my parents and siblings for their unconditional love, constant encouragements and occasional childcare services; my niece Yakout for brilliantly proving that humans as young as 8-month-old can break moving camouflage; Reda, my best friend and partner in life, for his endless love, support and patience; my daughter Aïda, born during this thesis, forever changing my life for the better.

Contents

1 Introduction and Background **9**

1.1 Motivation 10

1.2 Problem Definition and Background Literature 12

1.2.1 Motion Segmentation Approaches 13

1.2.2 Visually Challenging Objects 15

1.3 Thesis Outline and Contributions 19

1.3.1 Dataset Contributions 20

1.3.2 Publications 21

2 Betrayed by Motion: Camouflaged Object Discovery via Motion Segmentation **23**

2.1 Introduction 24

2.2 Related Work 27

2.3 A Video Registration and Segmentation Network 28

2.3.1 Motion Representation 29

2.3.2 Differentiable Registration Module 30

2.3.3 Motion Segmentation Module 32

2.4 MoCA: a new Moving Camouflaged Animal dataset 32

2.5 Experiments 34

2.6 Conclusions 41

Appendices **42**

2.A Homography Transformation Estimation 42

2.B Synthetic Moving Chairs 43

2.C End-to-end Motion Segmentation Pipeline 45

3 Segmenting Invisible Moving Objects **49**

3.1 Introduction 51

3.2 Related Work 53

3.3 Synthetic Data Generation 54

3.3.1 Foreground Objects 55

3.3.2 Generating Motion Sequences 55

3.3.3 Artificial Occluders 56

3.4 Architecture 57

3.5 Experiments 59

3.6 Results 61

3.7 Conclusions 65

Appendices **66**

3.A Synthetic Motion Simulation 66

3.B Dual-head Motion Segmentation Model 70

3.C Further Results 71

4 Self-supervised Video Object Segmentation by Motion Grouping **72**

4.1 Introduction 74

4.2 Related Work 76

4.3 Method 79

4.3.1 Flow Segmentation Architecture 80

4.3.2 Self-supervised Temporal Consistency Loss 83

4.4 Experimental Setup 85

4.5 Results . . . 87
4.6 Conclusions . . . 91

Appendices **92**

4.A Training Details . . . 92
4.B MoCA Dataset Curation . . . 93
4.C Further Results . . . 94

5 The Making and Breaking of Camouflage **97**

5.1 Introduction . . . 98
5.2 Related work . . . 100
5.3 Measuring the Effectiveness of Camouflage . . . 103
5.3.1 Perceptual Camouflage Score . . . 103
5.3.2 Probabilistic Scoring Function . . . 106
5.4 Generating Camouflage Video Sequences . . . 107
5.4.1 Camouflage Image Generation . . . 107
5.4.2 Camouflage Video Generation . . . 108
5.5 Learning to Break Camouflage . . . 109
5.6 Experiments . . . 110
5.7 Conclusions . . . 116

Appendices **117**

5.A Ranking Camouflage Images . . . 118
5.B Synthetic Camouflage Dataset . . . 126
5.C Moving Camouflaged Animal Segmentation . . . 128

6 Discussion **133**

6.1 Achievements and Impact . . . 133
6.2 Privacy and Ethical Responsibility . . . 135

6.3 Future Work 136

References **139**

A Statement of Authorship **154**

Chapter 1

Introduction and Background

The human visual system perceives objects: (i) by recognising distinctive visual features such as shapes, colours or textures; and/or (ii) by focusing on motion patterns that make them stand out from their background as they move. In some cases, appearance attributes are weak or non-informative: for instance, a camouflaged animal that blends perfectly with its background or an object that becomes occluded by another object or element in the scene, creating visual ambiguity. Motion cues are essential in perceiving such challenging objects. The camouflaged animal would stand out as it starts to move. Another example is that of transparent objects such as glass cups or soap bubbles that we can track simply by leveraging minimal cues of moving light reflections. Our visual system is even robust to occlusion as we are able to maintain object permanence, *i.e.* continuously tracking and perceiving objects as a whole even when they disappear behind an occluder.

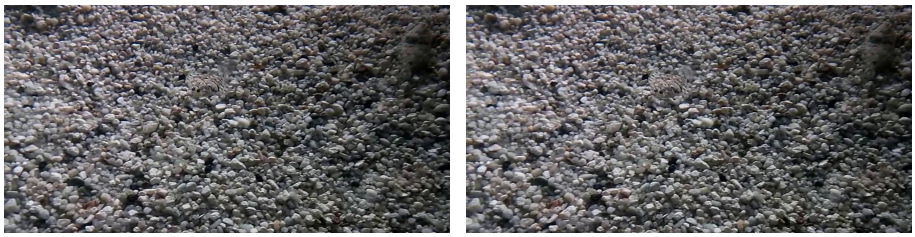


Figure 1.1: Example of an animal in camouflage: Given frame t (left) and frame $t+1$ (right), can you find the animal? (Answer provided in the next page. Do not cheat!)

In contrast, most existing methods for computational video object segmentation fail in these challenging scenarios as they rely primarily on visual appearance.

The goal of this thesis is to leverage motion as a key attribute to robustly segment objects of minimal visibility in videos.



Figure 1.2: Example of an animal in camouflage. Segmentation of frame t depicting a flatfish.

1.1 Motivation

Motion: a complementary cue to visual attributes

Motion plays a crucial role in human's perception and understanding of the world. In fact, we begin to attend to movement in our earliest interactions with the environment: generally, by the age of seven months, infants can impressively track rotating toys on mobiles hanging above their cradle [Pieh et al. 2012]. Upon maturation, our visual system would be able to perceive objects by relying on motion to disambiguate the visually challenging scenes. This understanding of objectness from motion precedes the ability to recognise distinctive appearance attributes (*e.g.* a particular shape or textural patterns).

Figure 1.3 describes an experiment where an 8-month-old infant was presented with a 2D object, a patch, camouflaged within a textured board. Incapable of seeing the object at first while it was static, the infant was able to perceive the object as soon as it started moving as shown in Figure 1.4.

How do we perceive such visually challenging objects? There has been evidence, in the neuroscience field [Goodale and Milner 1992], that the human visual system is segregated into two separate streams: the ventral stream responsible for processing the visual attributes of an object: shape, texture, colour, *etc.* and the dorsal stream in charge of processing motion. Performed simultaneously, this process allows for the motion to compensate for the lack of informative visual cues and vice versa. This dual stream theory has inspired numerous works and showed considerable success in action recognition and detection [Simonyan and Zisserman



Figure 1.3: **The moving camouflage experiment**: An 8-month-old infant is presented with a textured board and a detached patch camouflaged within a region of a similar texture to its own. First, the patch is static, then it undergoes a translational motion under the action of an invisible thread.

2014; El-Nouby and Taylor 2018] and video object segmentation [Tokmakov et al. 2017b; Tokmakov et al. 2019; Tianfei Zhou et al. 2020] by proposing two-pipeline architectures to process optical flow and RGB frames simultaneously. While these methods have achieved notable success in segmenting generic objects in videos, they struggle with the challenging cases such as camouflage, as they tend to overfit on the appearance stream while discarding important motion cues.



(a) Static phase (b) Motion phase (c) Motion phase

Figure 1.4: The infant acts indifferently when the object is static 1.4a. Then, as soon as the object starts moving, the infant begins to look at the object 1.4b, tracks it and finally grasps it 1.4c.

The psychology of vision

Another remarkable ability of the human visual system, is to perceive parts of occluded objects as a one entity, or associate elements moving in the same manner as parts of the same object. Psychologists have long studied these amazing perceptual abilities. Gestalt principles [Wertheimer 1923] were proposed as rules describing how the human visual system interprets these ambiguous cases by perceptual grouping. Among these laws: continuation, proximity, closure, similarity

and common fate. The latter principle states that elements moving in the same direction, with common velocity, seem to belong to the same entity. This principle inspired numerous works in the computer vision field [Mahendran et al. 2018; Tangemann et al. 2021].

1.2 Problem Definition and Background Literature

Motion segmentation aims at partitioning a sequence of frames into regions separating independently moving entities. These entities include a static or dynamic background and objects segmented based on their movement, in a class-agnostic manner. Considerable research works have been dedicated to this task, whether as a binary segmentation problem [Papazoglou and Ferrari 2013; Barnich and Van Droogenbroeck 2010; K. Kim et al. 2004], separating the background from independently moving objects, or the separate segmentation of multiple moving objects, by grouping pixels that undergo the same motion [Shen et al. 2018; Keuper et al. 2015; Brox and Malik 2010]. A thorough definition has been proposed in [Bideau and Learned-Miller 2016a] for both the binary and general motion segmentation tasks. The problem this thesis addresses is built on their definition of binary motion segmentation while taking into account the following requirements:

- Entirely segment partially moving objects: *e.g.* an animal that only moves the head should be segmented entirely, *i.e.* both its head and body should be segmented.
- Segment only freely moving objects: This means to discard the tethered or fixated elements in the scene. For instance, tree branches shaken by the wind or a flower whose petals are moving under the effect of a pollinating insect should not be segmented as moving objects.
- Segment momentarily static objects: We consider the notion of motion through the entire studied frame sequence and the segmentation should be maintained even when the object ceases to move in a sub-sequence.

Ambiguity with Unsupervised Video Object Segmentation (UVOS) Unsupervised Video Object Segmentation (UVOS) is a closely related task to motion segmentation, and sometimes they are referred to interchangeably in the literature. In UVOS, the goal is to segment objects in a video without any *a priori* information about the objects of interest (such as the groundtruth mask of the first frame). By the definition stated previously, motion segmentation is considered as a particular case for the general task of UVOS, that focuses on movement as a principal cue for video object discovery.

In the following, we review different approaches for motion segmentation and recall previous methods that tackled the specific case of visually challenging objects.

1.2.1 Motion Segmentation Approaches

We organise the main approaches for motion segmentation into three categories, although they might intersect as many works fall under multiple categories.

Clustering point trajectories

These methods formulated the problem of motion segmentation as clustering point trajectories or tracks derived from optical flow [Keuper et al. 2015; Brox and Malik 2010; Fragkiadaki et al. 2012; Keuper 2017; Ochs and Brox 2011; Yan and Pollefeys 2006; Shen et al. 2018; Ochs et al. 2013; F. Shi et al. 2013]. The clustering step has been addressed differently: graph-cut has been adopted in [Rahmati et al. 2014; Keuper et al. 2015; Keuper 2017] whereas [Brox and Malik 2010; Ochs et al. 2013; F. Shi et al. 2013] used spectral clustering [Ng et al. 2001]. However, they only produced sparse object labelling due to the complexity of detecting motion in homogeneous textureless regions and computational limitations. In [Ochs and Brox 2011], they further made an extension to resolve the problem of sparse labelling using superpixels. The fundamental drawback of these point track-based methods is that they assume that objects have a homogeneous translation motion and fail when this hypothesis does not hold. To address this, [Papazoglou and Ferrari 2013] proposed a method that does not require any hypothesis on the motion model or homogeneity of the object. Instead, they generate an initial segmentation mask using motion boundaries based on the optical flow, then use a mixture of Gaussians appearance model estimate over the RGB space to refine the labelling

through time. This method is again limited by the assumption of smoothness over time and requires the objects to have a significantly different motion from their backgrounds. Recently, [C. Xie et al. 2019] proposed a convolutional neural network that learns foreground trajectory embedding and discovers moving object by clustering using the Mises-Fisher mean shift (vMF- MS) algorithm [Kobayashi and Otsu 2010].

Geometric approaches

These methods tackle the motion segmentation task with an explicit geometrical representation of the scene. The *background subtraction* methods build a background model and use it to align the sequence frames so that the moving elements stand out in the difference of registered adjacent frames. In the case of stationary camera, which is common in video surveillance applications, the transformation mapping the sequence frames is the identity. This assumption was adopted by an extensive line of works: [Elgammal et al. 2000; Wren et al. 1997; Stauffer and Grimson 2000] focused on probabilistic modelling of pixel change such as intensity, while [Mittal and Paragios 2004; Pless et al. 2003; Sheikh and Shah 2005; Ko et al. 2010; V. Mahadevan and Vasconcelos 2008] adopted dynamic models predicting the current background based on past observations. *Projective geometry* [Hartley and Zisserman 2004] is used to compensate for the observer's motion in the case of a moving camera. For instance, the *layered representation* by [J. Y. Wang and Adelson 1994] decomposes a sequence into layers accumulated from all frames into a large map obtained by warping the individual frames using affine transformations. In other works, the transformation between adjacent images is approximated by homographies [Ren et al. 2003; Mittal and Huttenlocher 2000; Torr 1998]. The major drawback of these methods is that the homography approximation is only reasonable in the case of a distant background that can be considered planar and fails in the case of a more complex scene structure at closer distance.

Optical flow methods

Optical flow is considered a powerful signal for motion segmentation as it reflects change in terms of motion patterns within the scene through spatial mapping of video frames. Several works have exploited optical flow for motion segmentation,

either used as an input signal [W. Wang et al. 2018; Faktor and Irani 2014; Keuper et al. 2015; Papazoglou and Ferrari 2013; J. Cheng et al. 2017; Jain et al. 2017; Tokmakov et al. 2019; Song et al. 2018; Xiankai et al. 2019; Tianfei Zhou et al. 2020; S. Mahadevan et al. 2020; Y. Yang et al. 2019; Tianfei Zhou et al. 2020; Narayana et al. 2013] or as a supervision signal for self-supervised methods [Choudhury et al. 2022; Kipf et al. 2021]. Another line of works tackled the joint problem of simultaneously estimating optical flow and segmenting moving object in videos. An example of these methods addressed the problem using layered representations [Sun et al. 2013; Chang and Fisher 2013; B. Cheng et al. 2022]. More recently, SegFlow [J. Cheng et al. 2017] was proposed as a convolutional neural network that jointly learns to segment moving objects and estimate the optical flow of a given sequence. One of the limitations of using optical flow as an input signal is the so-called depth-dependence, *i.e.* objects sharing similar motion but placed at different distances from the camera, appear differently in the optical flow. For instance, elements of the background, even static, may appear to undergo different motions when placed at different depths, which can cause over-segmentation. A line of work addressed this problem and investigated solutions to disentangle flow and depth using flow orientation [Narayana et al. 2013; Bideau and Learned-Miller 2016b; Bideau et al. 2018].

A different line of work investigated the mutual information of optical flow regions [Y. Yang et al. 2019; Y. Yang et al. 2021], that is, if the optical flow of a given region cannot be predicted from the optical flow of its complementary region, then this indicates an independent motion. [Y. Yang et al. 2019] proposed a Generative Adversarial Networks setting (GAN), where a generator network outputs a mask defining a region and an inpainter attempts to predict its optical flow.

1.2.2 Visually Challenging Objects

When tracking moving objects, usual approaches focus on visual cues such as keypoints or distinctive visual features that differentiate the object from its background. However, when this information is weak or not available, visual tracking becomes highly challenging. In this thesis, we focus on different cases of challenging moving objects with the common aspect of limited visibility.

Camouflaged Animals

"In the arctic regions there are a number of animals which are permanently white or nearly so, or which only turn white in winter. (...) The obvious explanation of this style of colouration is, that it is protective, serving to conceal the herbivorous species from their enemies, and enabling carnivorous animals to approach their prey unperceived."
[Wallace 1889]

The *camouflage* phenomenon has been well studied in the biology field, with a rich history dating back to the seminal works of [Darwin 1859] and [Wallace 1889], as it offers great examples that illustrate the natural selection and adaptation. These fascinating animals have evolved to develop an appearance specifically designed to confuse and deceive the observer. Imitating patterns and textures from the background, they create strong visual ambiguity compared to regular objects.

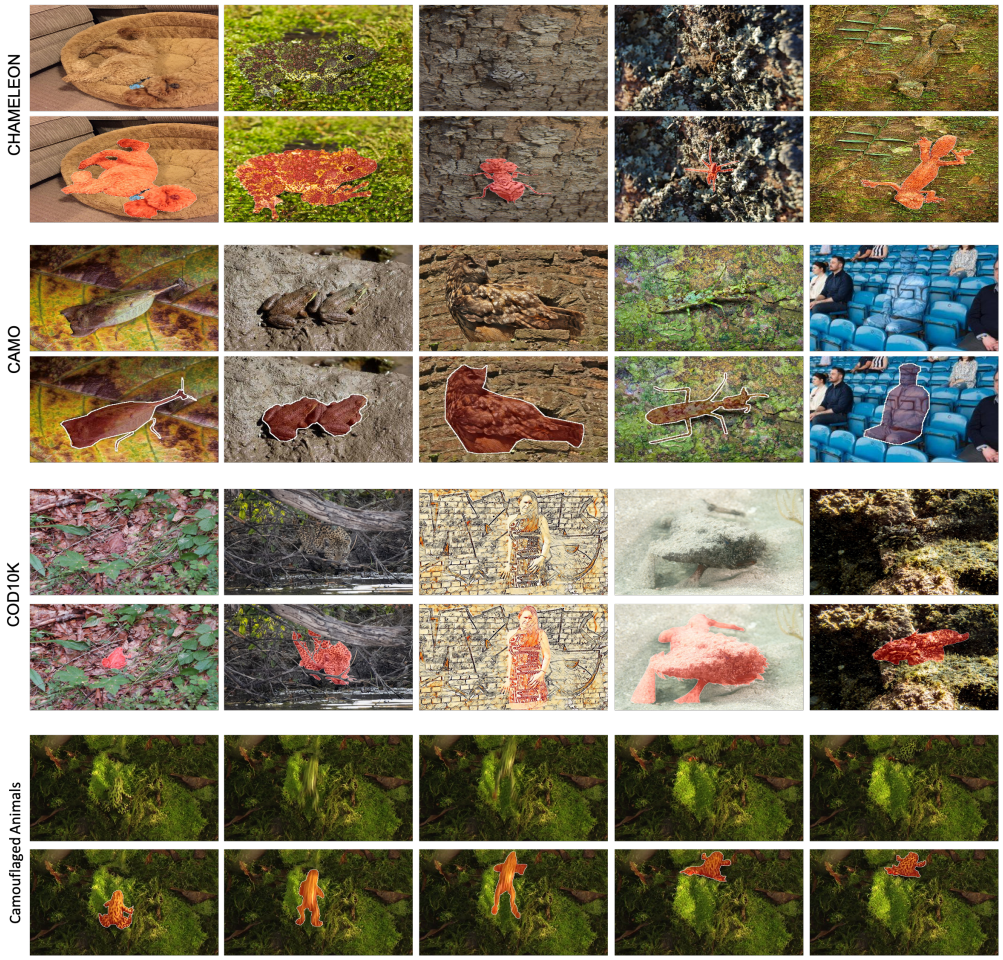


Figure 1.5: **Randomly sampled examples from different camouflage datasets**. Note that CHAMELEON [Soloveitchik et al. 2021], CAMO [Le et al. 2016] and COD10K [D. Fan et al. 2021] are still-image datasets, while Camouflaged Animals [Bideau and Learned-Miller 2016b] is a video dataset.

Multiple datasets have been proposed for segmenting camouflaged objects in still images, such as COD10K [D. Fan et al. 2021], CAMO [Le et al. 2016], CHAMELEON [Soloveitchik et al. 2021] and NC4K [Lv et al. 2021] (See Figure 1.5 for an illustration of randomly sampled examples from these datasets). This allowed the emergence of a line of methods that have addressed the problem from a single frame observation [Le et al. 2016; D. Fan et al. 2021; Lv et al. 2021; Q. Zhai et al. 2021; Ji et al. 2022; Mei et al. 2021]. However, these methods rely on visual cues such as boundaries which is counter-intuitive for camouflaged animals. A perfectly camouflaged object would completely blend with its background with an invisible contour and only becomes distinguishable from its surrounding as it starts to move. Hence, recent research direction has focused on motion cues to discover moving camouflaged animals in videos [Bideau and Learned-Miller 2016b; X. Cheng et al. 2022]. The work included in this thesis follows a similar direction.

Amodal Perception of Partially Occluded Objects

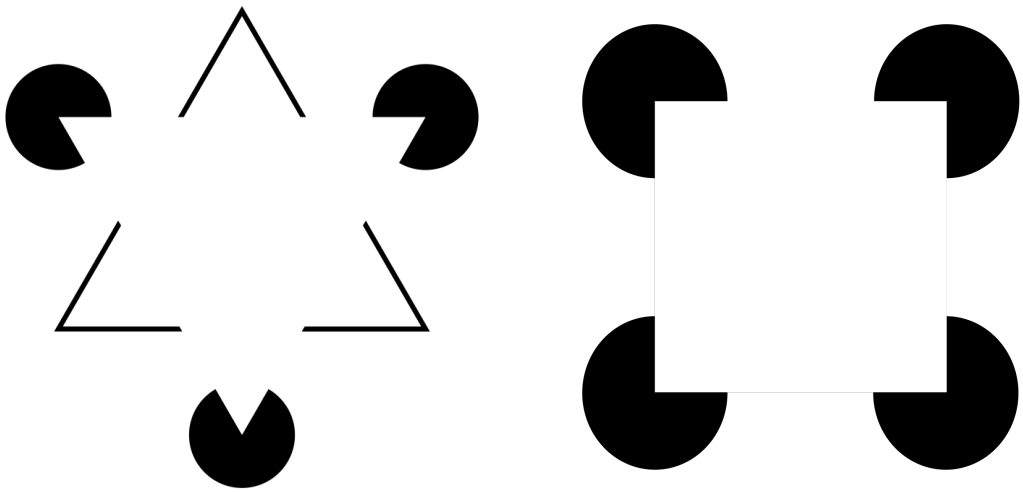


Figure 1.6: Illustration of amodal completion: In the Kanizsa triangle (left), V-shaped segments are perceived as a complete occluded black triangle outline behind a white triangle (resulting from the alignment of black Pacmen, perceived as three black filled circles). The Kanizsa square examples (right), the black Pacmen are perceived as four black filled circles occluded by a white square.

Our human visual system is capable of perceiving the full extent of object boundaries beyond occlusions, via complex contextual reasoning. This ability has been termed by psychologists as *amodal perception*. This is well illustrated by the Kanisza triangle and square Figure 1.6. However, the problem is far from be-

ing solved for computer vision, as occlusion remains one of the most challenging obstacles for the main tasks. In an effort to achieve this object-centric perception, the amodal segmentation task has been proposed, where the goal is to segment objects as a whole regardless of their occluded parts [Zhu et al. 2017; K. Li and Malik 2016; Purkait et al. 2019; J. Xie et al. 2022].

Transparent Objects

Whether made of glass, plastic or even trapped air within a thin film of liquid (also known as soap bubbles), transparent objects are involved in our daily activities; detecting and tracking them is essential especially for first-person vision applications. By definition, these objects transmit light waves, allowing the observer to see the background through them subject to distortion caused by refraction. This makes their appearance strongly dependant on the background, *e.g.*, a glass bottle would appear differently when placed in front of different backgrounds. Early methods tackled transparent object recognition by exploiting local features. For instance, [Fritz et al. 2009] proposed a transparent visual word model based on SIFT and LDA features. Another line of work addressed the problem by analytically exploiting the light distortion property of transparent objects. [Maeno et al. 2013] introduced the light distortion features LFD to classify transparent objects. [Y. Xu et al. 2015; Y. Xu et al. 2019] further proved the usefulness of these features for the segmentation task by computing the likelihood of a pixel belonging to a transparent object or Lambertian background. Lately, several works have

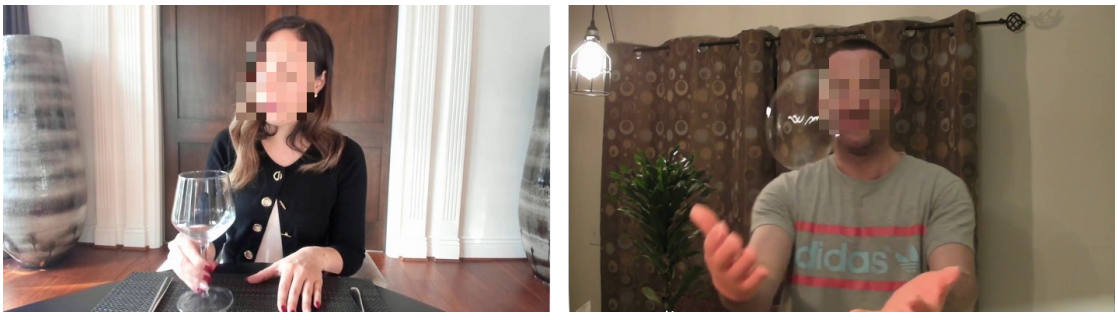


Figure 1.7: Examples from the Transparent Object Tracking benchmark (TOTB). Left: glassware, right: soap bubble. Note that faces have been anonymised in compliance with the EU General Data Protection Regulation (GDPR).

investigated the use of deep neural networks for segmenting transparent objects in single images [E. Xie et al. 2020; Kalra et al. 2020; Z. Zhou et al. 2020]. However, little research has been dedicated to segmenting moving transparent objects in

videos. Recently [H. Fan et al. 2021] proposed the Transparent Object Tracking Benchmark (TOTB), totalling 225 sequences. They revealed a significant gap in performance when analysing the state-of-the-art methods for tracking when applied to opaque objects compared to transparent objects. More recently, [Lukezic et al. 2022] further introduced a training dataset for transparent object tracking including over 2K sequences.

1.3 Thesis Outline and Contributions

In the following, we provide the outline of this thesis and a summary of its main contributions.

In chapter 2, we focus on the geometrical aspect of motion. Assuming independent motions of the foreground and background, we first compensate for camera motion by registering consecutive frames. We adopt the hypothesis that the background motion can be approximated by a homography transformation. To learn the parameters of this transformation, we introduce a differentiable registration module that operates in a RANSAC-like manner and train it on a synthetically generated dataset. We then use the registered image difference as a low-level signal that highlights the moving object and complements the optical flow by adding sharp object contours that may be lost in flow approximations. We observe that including the background registration leads to a significant gain in the motion segmentation performance. This chapter also introduces MoCA, the largest dataset for Moving Camouflaged Animal detection, that we use to test our motion segmentation pipelines in chapters 2 to 5.

In chapter 3, we address the challenging case of occluded objects and investigate a novel dual architecture that outputs both modal segmentation, indicating the visible part of the object, and the amodal segmentation, highlighting the object as a whole, even in the case of occlusion. Additionally, we introduce a scalable synthetic data generation pipeline while simulating various motion types and gradual object complexity and including occluders. To minimise the domain gap in comparison to real examples, we generate the synthetic sequences in the flow space. We found that our model trained on the synthetic dataset generalised well to real data and achieved state-of-the-art performance on MoCA at the time of publication.

In chapter 4, we tackle the task of motion segmentation from an object-centric angle. We build on slot-attention [Locatello et al. 2020] and train a neural network to perform a layered reconstruction of optical flow in a self-supervised manner. Our experiments demonstrate significant improvement on the MoCA dataset, as well as popular video object segmentation benchmarks such as DAVIS2016, outperforming self-supervised methods while remaining comparable to supervised methods that take both optical flow and RGB frames as input.

In chapter 5, we focus on generating synthetic camouflage images and video sequences while maintaining the visually challenging aspect of camouflage. To this end, we investigate perceptual score functions that quantify camouflage effectiveness and provide rankings of all publicly available camouflage datasets. We find that training a generative model with the camouflage effectiveness loss results in improving the generated camouflages in terms of blending with the background. We further generate synthetic videos of moving camouflaged animals that we use to train and improve our motion segmentation model.

In conclusion of this thesis, chapter 6 provides a discussion on the ethical responsibility and the impact of our research work and outlines possible future directions.

1.3.1 Dataset Contributions

The lack of appropriate data is one of the main challenges that we faced in this work. In fact, available video object segmentation benchmarks are not suited for the challenging cases that we address in this thesis. Additionally, manual annotation can be time-consuming, especially for camouflaged animals that often present complex shapes and thin structures taking up to 60 min per image [D. Fan et al. 2021]. Synthetic data generation offers a great alternative to circumvent real data scarcity with the major drawback of potential domain shift between the simulated data and the real world. Recent methods have shown success in closing this sim2real domain gap [J. Xie et al. 2022].

We summarise in the following the dataset contributions of our work:

MoCA: The Moving Camouflaged Animal dataset

We initially collected 141 videos of camouflaged animals from Youtube, totalling 37250 frames and 67 animal categories. We provided annotations in the form of bounding boxes as well as motion type (Locomotion, Deformation and Static). We further released a curated version by cropping channel logos and empty borders and keeping the predominant motion, resulting in 88 sequences and 4803 frames. The dataset is available for download at https://www.robots.ox.ac.uk/~vgg/data/MoCA/.

Synthetic datasets

For alignment We propose a data generation protocol that creates a sequence from a single background image by applying a sequence of homographies to randomly selected chairs. A detailed description of the generation process is provided in Appendix 2.B.

For modal/amodal motion segmentation We build a scalable data generation pipeline that creates sequences of optical flows from a background and an independently moving foreground consisting of polygons and real shapes and including occluders. This generated data provides both modal and amodal segmentation groudtruths. Further details are provided in Appendix 3.A.

For effective camouflage breaking We train a neural network in a generative adversarial setting, to create images of highly effective camouflage, together with the corresponding groundtruth segmentation, that we transform to sequences similarly to the previous dataset.

1.3.2 Publications

We present in chapters 2 to 4, the papers resulting from the research conducted in the course of this thesis, each of which has been peer-reviewed and accepted at a conference. The paper presented in chapter 5 is in the review process. With the exception of formatting, the papers have been left unmodified from their submitted versions. Appendix A provides authorship statements for each of these papers detailing the candidate's independent research contributions.

- Chapter 2: Betrayed by Motion: Camouflaged Object Discovery via Motion Segmentation

  Hala Lamdouar, Charig Yang, Weidi Xie, Andrew Zisserman

  Published in the proceedings of the Asian Conference on Computer Vision (ACCV), 2020.

- Chapter 3: Segmenting Invisible Moving Objects

  Hala Lamdouar, Weidi Xie, Andrew Zisserman

  Published in the proceedings of the British Machine Vision Conference (BMVC), 2021.

- Chapter 4: Self-supervised Video Object Segmentation by Motion Grouping

  Charig Yang, **Hala Lamdouar**, Erika Lu, Andrew Zisserman, Weidi Xie

  Published in the proceedings of the International Conference on Computer Vision (ICCV), 2021.

- Chapter 5: The Making and Breaking of Camouflage

  Hala Lamdouar, Weidi Xie, Andrew Zisserman

  Paper under review.

Chapter 2

Betrayed by Motion: Camouflaged Object Discovery via Motion Segmentation

This paper has been accepted for publication at the Asian Conference on Computer Vision (ACCV), 2020.

Betrayed by Motion: Camouflaged Object Discovery via Motion Segmentation

Hala Lamdouar Charig Yang Weidi Xie
Andrew Zisserman

Abstract

The objective of this paper is to design a computational architecture that discovers camouflaged objects in videos, specifically by exploiting motion information to perform object segmentation. We make the following three contributions: (i) We propose a novel architecture that consists of two essential components for breaking camouflage, namely, a differentiable registration module to align consecutive frames based on the background, which effectively emphasises the object boundary in the difference image, and a motion segmentation module with memory that discovers the moving objects, while maintaining the object permanence even when motion is absent at some point. (ii) We collect the first large-scale Moving Camouflaged Animals (MoCA) video dataset, which consists of over 140 clips across a diverse range of animals (67 categories). (iii) We demonstrate the effectiveness of the proposed model on MoCA, and achieve competitive performance on the unsupervised segmentation protocol on DAVIS2016 by only relying on motion.

2.1 Introduction

We consider a fun yet challenging problem of breaking animal camouflage by exploiting their motion. Thanks to years of evolution, animals have developed the ability to hide themselves in the surrounding environment to prevent being noticed by their prey or predators. Consider the example in Figure 2.1a, discovering the

fish by its appearance can sometimes be extremely challenging, as the animal's texture is indistinguishable from its background environment. However, when the fish starts moving, even very subtly, it becomes apparent from the motion, as shown in Figure 2.1c. Having the ability to segment objects both in still images, where this is possible, and also from motion, matches well to the two-stream hypothesis in neuroscience. This hypothesis suggests that the human visual cortex consists of two different processing streams: the ventral stream that performs recognition, and the dorsal stream that processes motion [Goodale and Milner 1992], providing strong cues for visual attention and detecting salient objects in the scene.



(a) t-th frame. (b) (t+1)-th frame. (c) Optical flow.

Figure 2.1: Two consecutive frames from the camouflage dataset, with a bounding box denoting the salient object. When the object starts moving, even subtly, we are able to detect it more easily, as shown, in the computed optical flow.

In recent years, computer vision research has witnessed tremendous progress, mainly driven by the ability of learning effective representations for detecting, segmenting and classifying objects in *still images*. However, the assumption that objects can be well-segmented by their appearance alone is clearly an oversimplification; that is to say, if we draw an analogy from the two-stream hypothesis, the computer vision systems trained on images can only mimic the function of the ventral stream. The goal of this paper is to develop a computational architecture that is able to process motion representations for breaking camouflage, *e.g.* by taking optical flow (Figure 2.1c) as input, and predicting a segmentation mask for the animal of interest.

Unfortunately, simply relying on motion will not solve our problem completely, as, *first*, optical flow estimation itself remains extremely challenging and is under active research. In practice, modern optical flow estimation techniques provide a fairly good indication of rough object motion, but not fine-grained details, *e.g.* the exact shape of the objects and their contours. To compensate for the missing details, we propose to use a differentiable registration module for aligning consecutive

frames, and use the difference of the registered images as auxiliary information to determine the exact contour; *Second*, if the motion stops at certain points in the video sequence, then a memory module is required to maintain the object permanence, as is done in [Tokmakov et al. 2019], *i.e.* to capture the idea that objects continue to exist even when they cannot be seen explicitly in the motion representation.

Another main obstacle encountered when addressing the challenging task of camouflage breaking is the limited availability of benchmarks to measure progress. In the literature, there is a Camouflaged Animals video dataset, released by Bideau *et al.* [Bideau and Learned-Miller 2016a], but this has only 9 clips on 6 different kinds of animals (about 840 frames). To overcome this limitation, we collect a **Mo**ving **C**amouflaged **A**nimal dataset, termed **MoCA**, which consists of 141 video sequences (37K frames), depicting 67 kinds of camouflaged animals moving in natural scenes. Both temporal and spatial annotations are provided in the form of tight bounding boxes on every 5th frame and the rest are linearly interpolated.

To summarize, in this paper, we make the following contributions: *First*, we propose a novel architecture with two essential components for breaking camouflage, namely, a differentiable registration module to align the background (regions other than the camouflaged animal) of consecutive frames, which effectively highlights the object boundary in the difference image, and a motion segmentation module with memory that discovers moving objects, while maintaining the object permanence even when motion is absent at some point. *Second*, we collect the first large-scale video camouflage benchmark (MoCA), which we release to the community for measuring progress on camouflage breaking and object tracking. *Third*, we demonstrate the effectiveness of the proposed model on MoCA, outperforming the previous video segmentation approaches using motion. In addition, we also benchmark on DAVIS2016, achieving competitive performance on the unsupervised segmentation protocol despite using only motion. Note that, DAVIS2016 is fundamentally different from MoCA, in the sense that the objects are visually distinctive from the background, and hence motion may not be the most informative cue for segmentation.

2.2 Related Work

Our work cuts across several areas of research with a rich literature, we can only afford to review some of them here.

Video Object Segmentation [Bideau and Learned-Miller 2016a; Pont-Tuset et al. 2017; N. Xu et al. 2018; Brox and Malik 2010; Ochs and Brox 2011; Papazoglou and Ferrari 2013; Jain et al. 2017; Tokmakov et al. 2019; Dave et al. 2019; Oh et al. 2019; Voigtlaender et al. 2019; Vondrick et al. 2018; J. Wang et al. 2019; Lai and W. Xie 2019; Lai et al. 2020; Maninis et al. 2018; Voigtlaender and Leibe 2017; Caelles et al. 2017; Fragkiadaki et al. 2012; Keuper et al. 2015; Z. Yang et al. 2019; Xiankai et al. 2019] refers to the task of localizing objects in videos with pixel-wise masks. In general, two protocols have recently attracted an increasing interest from the vision community [Pont-Tuset et al. 2017; N. Xu et al. 2018], namely unsupervised video object segmentation (**unsupervised VOS**), and semi-supervised video object segmentation (**semi-supervised VOS**). The former aims to automatically separate the object of interest (usually the most salient one) from its background in a video sequence; and the latter aims to re-localize one or multiple targets that are specified in the first frame of a video with pixel-wise masks. The popular methods to address the unsupervised VOS have extensively relied on a combination of appearance and motion cues, *e.g.* by clustering trajectories [Brox and Malik 2010; Ochs and Brox 2011], or by using two stream networks [Papazoglou and Ferrari 2013; Jain et al. 2017; Tokmakov et al. 2019; Dave et al. 2019]; or have purely used appearance [J. Wang et al. 2019; Z. Yang et al. 2019; Jun Koh and C. Kim 2017; D. Fan et al. 2019]. For semi-supervised VOS, prior works can roughly be divided into two categories, one is based on mask propagation [Oh et al. 2019; Voigtlaender et al. 2019; Vondrick et al. 2018; J. Wang et al. 2019; Lai and W. Xie 2019; Lai et al. 2020], and the other is related to few shot learning or online adaptation [Maninis et al. 2018; Voigtlaender and Leibe 2017; Caelles et al. 2017].

Camouflage Breaking [Bideau and Learned-Miller 2016a; Le et al. 2016] is closely related to the unsupervised VOS, however, it poses an extra challenge,

as the object's appearance from *still image* can rarely provide any evidence for segmentation, *e.g.* boundaries. As such, the objects or animals will only be apparent when they start to move. In this paper, we are specifically interested in this type of problem, *i.e.* breaking the camouflage in a class-agnostic manner, where the model takes no prior knowledge of the object's category, shape or location, and is asked to discover the animal with pixel-wise segmentation masks whenever they move.

Image Registration/Alignment is a long-standing vision problem with the goal of transferring one image to another with as many pixels in correspondence as possible. It has been applied to numerous applications such as video stabilization, summarization, and the creation of panoramic mosaics. A comprehensive review can be found in [Hartley and Zisserman 2004; Szeliski 2004]. In general, the pipeline usually involves both correspondence estimation and transformation estimation. Traditionally, the alignment methods apply hand-crafted features, *e.g.* SIFT [Lowe 1999], for keypoint detection and matching in a pair of images, and then compute the transformation matrix by solving a linear system. To increase the robustness of the geometric transformation estimation, RANdom SAmple Consensus (RANSAC) [Fischler and Bolles 1981] is often adopted. In the deep learning era, researchers have constructed differentiable architectures that enable end-to-end optimization for the entire pipeline. For instance, [Brachmann et al. 2017] proposed a differentiable RANSAC by relaxing the sparse selection with a soft-argmax operation. Another idea is to train a network with binary classifications on the inliers/outliers correspondences using either ground truth supervision or a soft inlier count loss, as in [Brachmann and Rother 2018; Ranftl and Koltun 2018; Rocco et al. 2018; Brachmann and Rother 2019], and solve the linear system with weighted least squares.

2.3 A Video Registration and Segmentation Network

At a high level, we propose a novel computational architecture for breaking animal camouflage, which *only* considers motion representation as input, *e.g.* optical

flow, and produces the segmentation mask for the moving objects. Specifically, as shown in Figure 2.2, the model consists of two modules: (i) a differentiable registration module for aligning consecutive frames, and computing the *difference* image to highlight the fine-grained detail of the moving objects, *e.g.* contours and boundaries (Section 2.3.2); and (ii) a motion segmentation network, which takes optical flow together with the difference image from the registration as input, and produces a detailed segmentation mask (Section 2.3.3).

2.3.1 Motion Representation

In this paper, we utilise optical flow as a representation of motion. Formally, consider two frames in a video sequence, I_t and I_{t+1}, each of dimension $\mathbb{R}^{H\times W\times 3}$, the optical flow is defined as the displacement field $F_{t\to t+1} \in \mathbb{R}^{H\times W\times 2}$ that maps each pixel from I_t to a corresponding one in I_{t+1}, such that:

$$I_t(\mathbf{x}) = I_{t+1}(\mathbf{x} + F_{t\to t+1}(\mathbf{x})), \tag{2.1}$$

where $\mathbf{x}$ represents the spatial coordinates (x, y) and F represents the vector flow field in both horizontal and vertical directions. In practice, we use the pretrained PWCNet [Sun et al. 2018] for flow estimation, some qualitative examples can be found in Figure 2.1 and Figure 2.5.

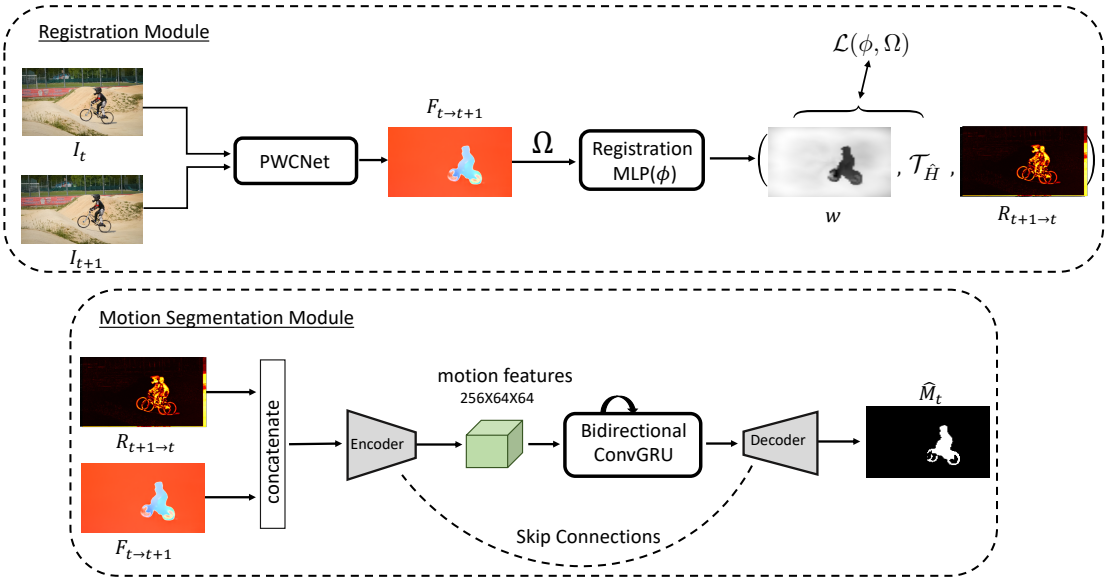


Figure 2.2: Architecture Overview. The proposed architecture is composed of two different modules, namely registration and motion segmentation.

2.3.2 Differentiable Registration Module

One of the main challenges of segmentation with optical flow is the loss of rich fine-grained details due to motion approximations. In order to recover the sharp contours of the objects under motion, we seek a low level RGB signal which ideally suppresses the background and highlights the object's boundaries. In this paper, we use the image difference between consecutive frames after camera motion compensation. To this end, a reasonable assumption to make is that the foreground object undergoes an independent motion with respect to the global transformation of the background. We propose a differentiable registration module for estimating this transformation, which we approximate with a homography ($\mathcal{T}_{\hat{H}}$) between consecutive frames, and then compute the *difference* image after alignment ($R_{t+1\to t} = |\mathcal{T}_{\hat{H}}(I_{t+1}) - I_t|$), which will provide cues for the animal contours.

The key here is to train a registration module that accepts a set of correspondences obtained from the consecutive frames, outputs an homography transformation matrix (H), and an inlier weight map (w), which, ideally, acts like a RANSAC process, and has 1's for every background pixel, and 0's for every foreground pixel (moving object's). In this paper, we parametrize the registration module with Multiple Layer Perceptrons (MLPs), *i.e.* $[H, w] = \phi(\{p_s, p_t\}; \theta_r)$, where $p_s \in \mathcal{R}^{mn\times 2}$ denotes the spatial coordinates of all pixels (normalized within the range $[-1, 1]$) in the source image, and their corresponding position in the target image ($p_t \in \mathcal{R}^{mn\times 2}$), based on the computed optical flow, and θ_r are the trainable parameters.

Homography Transformation.

In order to be self-contained, we summarise here the homography computation. Mathematically, a homography transformation ($\mathcal{T}_H$) maps a subset of points $\mathcal{S}_s$ from the source image to a subset of points $\mathcal{S}_t$ in the target image; in our case, the source and target images refer to I_{t+1} and I_t respectively:

$$\forall {p_i}^s \in \mathcal{S}_s,\ \exists {p_i}^t \in \mathcal{S}_t \quad {p_i}^t = \mathcal{T}_H({p_i}^s) = \alpha_i H {p_i}^s, \tag{2.2}$$

where H is the matrix associated with the homography transformation $\mathcal{T}_H$ with 8 degrees of freedom, and α_i a non-zero scalar. This formulation can be expressed

using homogeneous coordinates of ${p_i}^s$ and ${p_i}^t$ as:

$$\begin{pmatrix} {x_i}^t \\ {y_i}^t \\ 1 \end{pmatrix} = \alpha_i H \begin{pmatrix} {x_i}^s \\ {y_i}^s \\ 1 \end{pmatrix}. \tag{2.3}$$

Using the standard Direct Linear Transform (DLT) [Hartley and Zisserman 2004], the previous equation can be written as:

$$A\ vec(H) = \mathbf{0}, \tag{2.4}$$

where $vec(H) = \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{23} & h_{31} & h_{32} & h_{33} \end{pmatrix}^T$ is the vectorised homography matrix and A the data matrix. The homography H can therefore be estimated by solving such over-complete linear equation system. For more details on the DLT computation, refer to the extended ArXiv version of this paper.

Training objective.

In order to train the registration module ($\phi(\cdot;\theta_r)$), we can optimize:

$$\mathcal{L} = \frac{1}{\sum w} \sum_{\Omega} w \cdot ||\mathcal{T}_H(p_s) - p_t||_2 + R(w), \tag{2.5}$$

where Ω refers to all the pixels on the mn grid. Note that, the homography $\mathcal{T}_H$ transformation in this case can be solved with a simple weighted least square (WLS) and differentiable SVD [Ranftl and Koltun 2018] for parameter updating. To avoid trivial solution, where the weight map can be full of zeros that perfectly minimize the loss, we add a regularization term ($R(w)$), that effectively encourages as many inliers as possible:

$$R(w) = -\gamma \sum_{p \in \Omega} l_p - \frac{1}{mn} \sum_{\Omega} \Big\{ l_p \cdot \log(w) + (1 - l_p) \cdot \log(1 - w) \Big\}, \tag{2.6}$$
$$\text{where} \quad l_p = \sigma\{(\epsilon - ||\mathcal{T}_H(p_s) - p_t||_2)/\tau\}.$$

In our training, $\gamma = 0.05$, $\tau = 0.01$, $\epsilon = 0.01$ and $m = n = 64$. The first term in $R(w)$ (l_p) refers to a differentiable inlier counting [Brachmann and Rother 2018; Rocco et al. 2018]. The rest of the terms aim to minimize the binary cross-entropy at each location of the inlier map, as in [Moo Yi et al. 2018], forcing the predictions

to be classified as inlier (1's) or outlier (0's).

2.3.3 Motion Segmentation Module

After introducing the motion representation and registration, we consider a sequence of frames from the video, $I \in \mathcal{R}^{T\times 3\times H\times W}$, where the three channels refer to a concatenation of the flow (2 channels) and difference image (1 channel). For simplicity, here we use a variant of UNet [Ronneberger et al. 2015] with the bottleneck feature maps being recurrently processed.

Specifically, the **Encoder** of the segmentation module will independently process the current inputs, ending up with motion features of $\mathcal{R}^{T\times 256\times 64\times 64}$, where $T, 256, 64, 64$ refer to the number of frames, number of channels, height, and width respectively. After the Encoder, the **memory module** (a bidirectional convGRU [Ballas et al. 2016] is used in our case) operates on the motion features, updating them by aggregating the information from time steps in both directions. The **Decoder** takes the updated motion features, and produces an output binary segmentation mask, *i.e.* foreground vs background. The Motion Segmentation Module is trained with pixelwise binary cross-entropy loss.

This completes the descriptions of the two individual modules used in the proposed architectures. Note that the entire model is trained together as it is end-to-end differentiable.

2.4 MoCA: a new Moving Camouflaged Animal dataset

One of the main obstacles encountered when addressing the challenging task of camouflage breaking is the limited availability of datasets. A comparison of existing datasets in Table 2.1. Bideau *et al.* published the first Camouflaged Animals video dataset with only 9 clips, and Le *et al.* proposed the CAMO dataset with only single image camouflage, and therefore not suitable for our video motion segmentation problem.

To overcome this limitation, we collect the **Mo**ving **C**amouflaged **A**nimal dataset, termed **MoCA**, which consists of 141 video sequences depicting various camou-

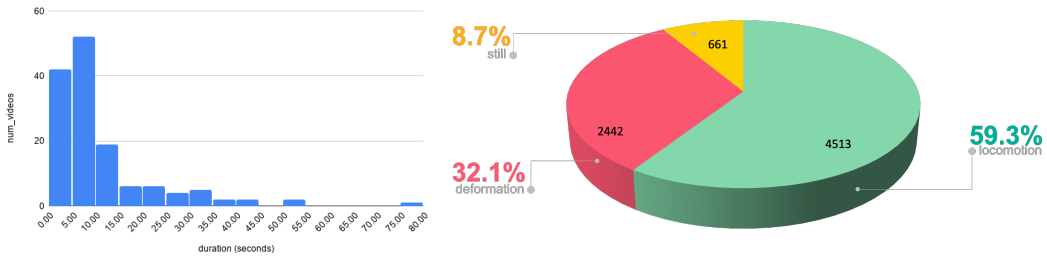
Table 2.1: Statistics for recent camouflage breaking datasets, namely Camouflaged Animals [Bideau and Learned-Miller 2016b] and CAMO [Le et al. 2016] compared to ours; "# Clips" denotes the number of clips in the dataset; "# Images" denotes the number of frames or images in the dataset; "# Animals" denotes the number of different animal categories in the dataset; "Video" indicates whether videos are available.

Datasets	# Clips	# Images	# Animals	Video
Camouflaged Animals	9	839	6	✓
CAMO	0	1250	80	×
MoCA (Ours)	141	37K	67	✓

flaged animals moving in natural scenes. We include the PWC-Net optical flow for each frame and provide both spatial annotations in the form of bounding boxes and motion labels every 5-th frame with the rest linearly interpolated.

2.4.1 Detailed statistics

MoCA contains 141 video sequences collected from YouTube, at the highest available resolution, mainly 720×1280, and sampled at $24fps$. A total of 37250 frames spanning 26 minutes. Each video represents a continuous sequence depicting one camouflaged animal, ranging from 1.0 to 79.0 seconds. The distribution of the



(a) Distribution of video lengths. (b) Distribution of motion labels.

Figure 2.3: Statistics for the Moving Camouflaged Animals (MoCA) dataset. In (a), the x-axis denotes the video duration, and the y-axis denotes the number of video sequences. (b) the distribution of frames according to their motion types (still, deformation and locomotion), see text for the detailed definitions.

video lengths is shown in Figure 2.3a. The dataset is labelled with a bounding box in each frame, as well as a motion label for the type of motion. While annotating the data, we distinguish three types of motion:

- **Locomotion:** when the animal engages in a movement that leads to a significant change of its location within the scene *e.g.* walking, running, climbing, flying, crawling, slithering, swimming, *etc.*

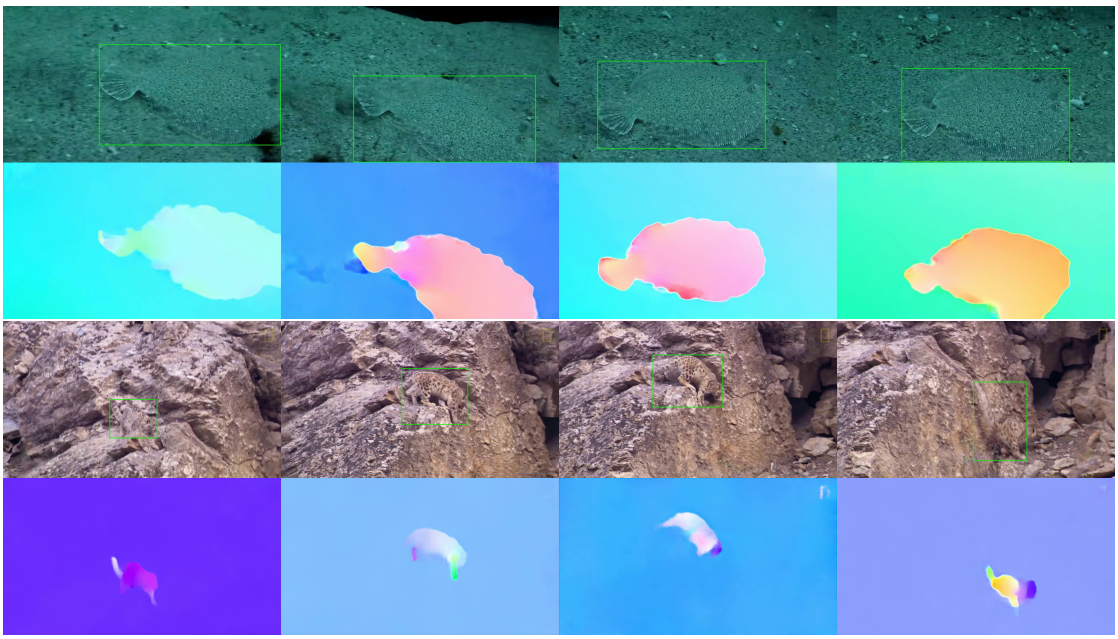


Figure 2.4: Example sequences from the Moving Camouflaged Animals (MoCA) dataset with their corresponding optical flows.

- **Deformation:** when the animal engages in a more delicate movement that only leads to a change in its pose while remaining in the same location *e.g.* moving a part of its body.

- **Static:** when the animal remains still.

As shown in Figure 2.3b, motion wise, the dataset contains 59.3% locomotion, 32.1% deformations, and 8.7% still frames. Examples from the dataset are shown in Figure 2.4.

2.5 Experiments

In this section, we detail the experimental setting used in this paper, including the datasets, evaluation metrics, baseline approaches and training details.

2.5.1 Datasets

DAVIS2016 refers to the Densely Annotated VIdeo Segmentation dataset [Perazzi et al. 2016]. It consists of 50 sequences, 30 for training and 20 for testing, captured at $24fps$ and provided at two resolutions. We use the $480p$ version in all experiments. This dataset has the advantage that accurate pixelwise ground truth segmentations are provided, $3,455$ annotations in total, as well as spanning

a variety of challenges, such as occlusions, fast-motion, non-linear deformation and motion-blur. We train the model on the DAVIS 2016 training set and report results on its validation splits.

Synthetic Moving Chairs. In order to train the differentiable registration module properly, we use video sequences that are synthetically generated. Specifically, we use the 3D-rendered objects from the Flying Chairs dataset as foreground, and take random images from YouTube-VOS as background. We then apply rigid motions, *e.g.* homographies, to the background, and simulate an independent motion for the foreground object. Note that, with such a synthetic dataset, we have complete information about the background homography transformation, optical flow, inlier maps, and object masks, which enables us to better initialise the registration module before training on real video sequences. These synthetic video sequences were only used to pre-train the registration and motion segmentation modules. The extended ArXiv version of this paper includes example images.

Evaluation Metrics. Depending on the benchmark dataset used, we consider two different evaluation metrics. For **DAVIS2016**, we follow the standard protocol for unsupervised video object segmentation proposed in [Perazzi et al. 2016], namely, the mean region similarity $\mathcal{J}$, which is the intersection-over-union (IOU) of the prediction and ground truth; and mean contour accuracy $\mathcal{F}$, which is the F-measure defined on contour points from the prediction and the ground truth. For **MoCA**, as we only have the bounding box annotations, we define the metric as the IOU between the groundtruth box and the minimum box that includes the predicted segmentation mask. Note that, we follow the same protocol used in Bideau *et al.* [Bideau and Learned-Miller 2016a], meaning, we only evaluate the segmentation of the animals under locomotion or deformation (*not* in static frames).

2.5.2 Baselines

We compare with five previous state-of-the-art approaches [Tokmakov et al. 2017a; Tokmakov et al. 2019; Dave et al. 2019; Z. Yang et al. 2019; Xiankai et al. 2019]. In [Tokmakov et al. 2019], Tokmakov *et al.*, the LVO method uses a two-stream

network for motion segmentation, where the motion stream accepts optical flow as input via a MPNet [Tokmakov et al. 2017a] architecture, and the appearance stream uses an RGB image as input. Similar to ours, a memory module is also applied to recurrently process the frames. A more recent approach [Dave et al. 2019] adapts the Mask-RCNN architecture to motion segmentation, by leveraging motion cues from optical flow as a bottom-up signal for separating objects from each other, and combines this with appearance evidence for capturing the full objects. For fair comparison across methods, we use the same optical flow computed from PWCNet for all the flow-based methods. To this end, we re-implement the original MPNet [Tokmakov et al. 2017a], and train on the synthetic FT3D dataset [Mayer et al. 2016]. For LVO [Tokmakov et al. 2019], we also re-train the pipeline on DAVIS2016. For Seg-det [Dave et al. 2019] and Seg-track [Dave et al. 2019], we directly replace the flows from Flownet2 [Ilg et al. 2017] with the ones from PWC-Net in the model provided by the authors. In all cases, our re-implemented models outperform or match the performance reported in the original papers. In addition, we also compare our method to AnchorDiff [Z. Yang et al. 2019] and COSNet [Xiankai et al. 2019], both approaches have been trained for unsupervised video object segmentation with only RGB video clips, and show very strong performance on DAVIS.

2.5.3 Training and Architecture Details

Registration Module. We adopt the architecture of [Moo Yi et al. 2018], which is MLPs with 12 layer residual blocks. We first train the registration module on the Synthetic Moving Chairs sequences described in 2.5.1, for $10K$ iterations using an Adam optimizer with a weight decay of 0.005 and a batch size of 4. For a more stable training, we use a lower learning rate, *i.e.* 5×10^{-5}, avoiding the ill-conditioned matrix in the SVD.

Motion Segmentation Module. We adopt a randomly initialized ResNet-18. Frame-wise segmentation is trained from scratch on the synthetic dataset, together with the pre-trained registration module. We further include the bidirectional ConvGRU and finetune the whole pipeline on DAVIS 2016, with each sequence of length 11, batch size of 2, for a total of $25K$ iterations. For all trainings, we use

frames with a resolution of $\mathcal{R}^{256\times256\times3}$.

2.5.4 Results

In this section, we first describe the performance of our model and previous state-of-the-art approaches on the new MoCA dataset, and then compare segmentation performance on the DAVIS2016 benchmark.

Table 2.2: Mean Intersection Over Union on MoCA for the different motion type subsets. "All_Motion" refers to the overall performance. We compare our methods against our method against MPNet[Tokmakov et al. 2017a], AD[Z. Yang et al. 2019], COSNet[Xiankai et al. 2019], LVO[Tokmakov et al. 2019], Seg-det[Dave et al. 2019] and Seg-track[Dave et al. 2019].

Input	Model	RGB	Flow	Register	Memory	Locomotion	Deform	All_Motion
Flow	MPNet	×	✓	×	✓	21.3	**23.5**	22.2
	ours-A	×	✓	×	✓	31.3	17.8	27.6
	ours-B	×	✓	MLP	×	29.9	15.6	25.5
	ours-C	×	✓	MLP	✓	**47.8**	20.7	**39.4**
	ours-D	×	✓	RANSAC	✓	42.9	19.2	35.8
RGB	AD	✓	×	×	×	30.9	29.4	30.4
	COSNet	✓	×	×	×	35.9	35.1	36.2
Both	LVO	✓	✓	×	✓	30.6	34.9	30.6
	Seg-det	✓	✓	×	×	16.9	18.7	17.9
	Seg-track	✓	✓	×	×	29.9	32.2	30.2
	ours-E	✓	✓	MLP	✓	**45.0**	**38.0**	**42.4**

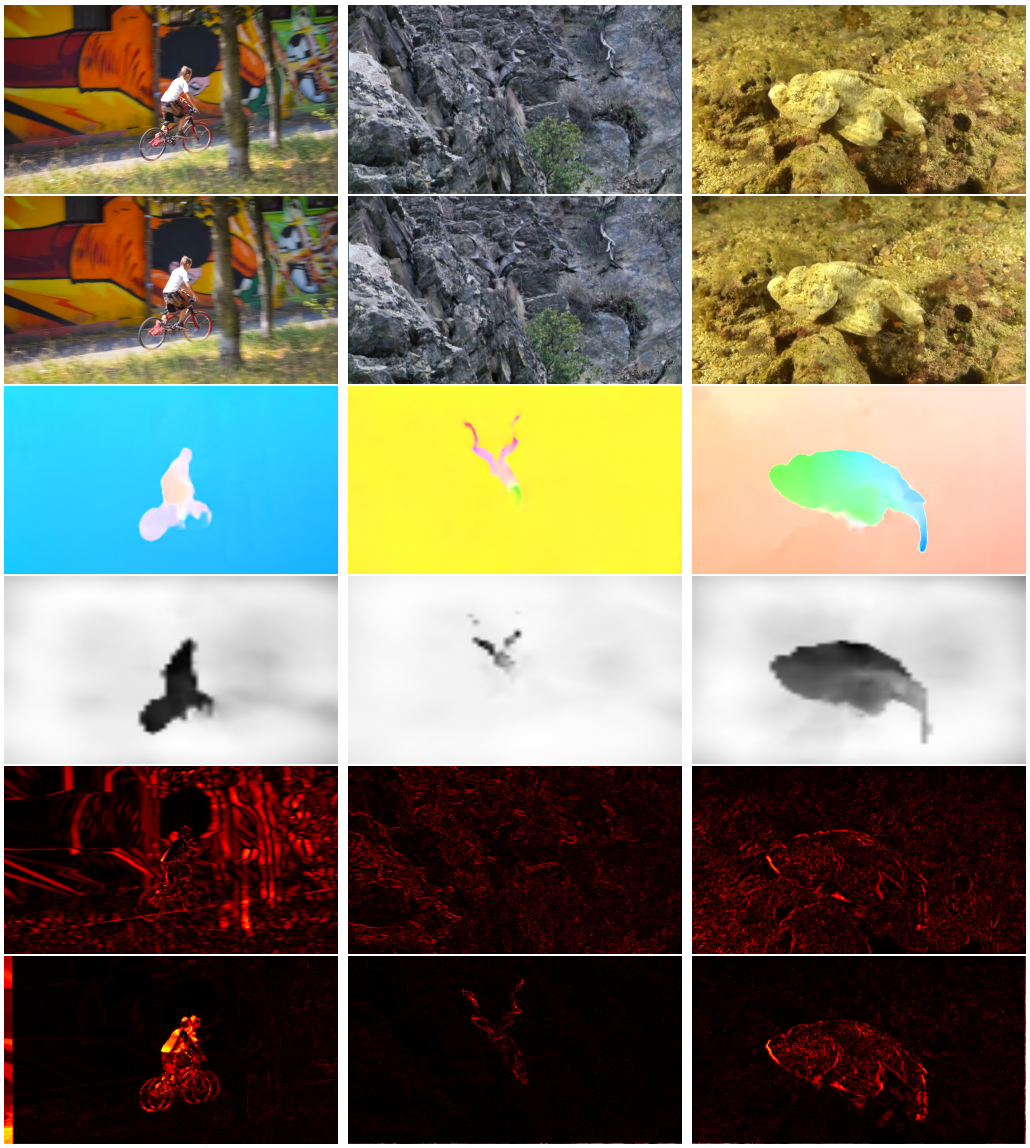
2.5.5 Results on the MoCA Benchmark

We summarise all the quantitative results in Table 2.2 and discuss them in the following sections. In Figure 2.5, we illustrate the effect of the differentiable registration and Figure 2.C.1 shows examples of the overall segmentation method.

Effectiveness of registration/alignment. To demonstrate the usefulness of the differentiable registration module, we carry out an ablation study. By comparing ours-A (without registration), ours-D (registration using RANSAC) and Ours-C (registration using our trainable MLPs), it is clear that the model with trainable MLPs for alignment helps to improve both the animal discovery on video sequences with locomotion and deformation, outperforming ours-A (without registration) and ours-D (RANSAC).

In Figure 2.5 we visualise the results from the registration module, *e.g.* the inlier map, difference image before alignment (second last row) and after alignment (last

row). It is clear that the difference images computed after alignment are able to ignore the background, and successfully highlight the boundary of the moving objects, *e.g.* the wheels of the bicycle from the first column.



(a) (b) (c)

Figure 2.5: Registration results from (a) the validation set of DAVIS 2016; and (b)(c) MoCA. From top to bottom: Frame t, Frame $t+1$, Forward PWCNet optical flow, inlier weights (background pixels are shown as gray or white), image difference without alignment, aligned image difference.

Effectiveness of memory. Comparing model-B (without memory) and model-C (with memory), the only difference lies on whether the frames are processed individually or recurrently. As shown by the results, a significant boost is obtained with the help of the memory module (25.5 vs. 39.4 on All_Motion), showing its effectiveness.

Comparison to baselines and previous approaches.

From Table 2.2, we make the following observations: *First*, when comparing with MPNet [Tokmakov et al. 2017a], which also processes optical flow, our model demonstrates a superior performance on All_Motion (39.4 vs 22.2), and an even larger gap on Locomotion, showing the usefulness of image registration and memory modules; *Second*, as expected, the state-of-the-art unsupervised video segmentation approaches relying on appearance (RGB image as input), *e.g.* AnchorDiff [Z. Yang et al. 2019] and COSNet [Xiankai et al. 2019], tend to struggle on this camouflage breaking task, as the animals often blend with the background, and appearance then does not provide informative cues for segmentation, emphasising the importance of motion information (by design) in this dataset; *Third*, when we adopt a two-stream model, *i.e.* extend the architecture with a Deeplabv3-based appearance model, and naively average the prediction from appearance and flow models, the performance can be further boosted from 39.4 to 42.4, significantly outperforming all the other two-stream competitors, *e.g.* LVO [Tokmakov et al. 2019], Seg-det and Seg-seg [Dave et al. 2019].

2.5.6 Results on DAVIS2016 Benchmark

We compare to previous approaches on the Unsupervised Video Object Segmentation protocol. In all experiments, we *do not* use any post-processing *e.g.* CRF. Both our re-trained MPNet and LVO model outperform the original reported results in the papers, which guarantees a fair comparison with their model. For Seg-det, replacing the PWC-Flow leads to a small drop (around 2.3%) in the mean $\mathcal{J}$, but this will not affect our conclusion here.

As shown in Table 2.3, when compared with MPNet which also take flow-only input, our model (ours-C) outperforms it on all metrics by a large margin. Note that, the DAVIS benchmark is fundamentally different from MoCA, as the approaches only relying on appearance are very effective [Z. Yang et al. 2019], indicating that the objects in the DAVIS sequences can indeed be well-identified by the appearance, and motion is not playing the dominant role as it is in MoCA. This can also be observed from the results for Seg-det, Seg-track, AnchorDiff and COSNet, which

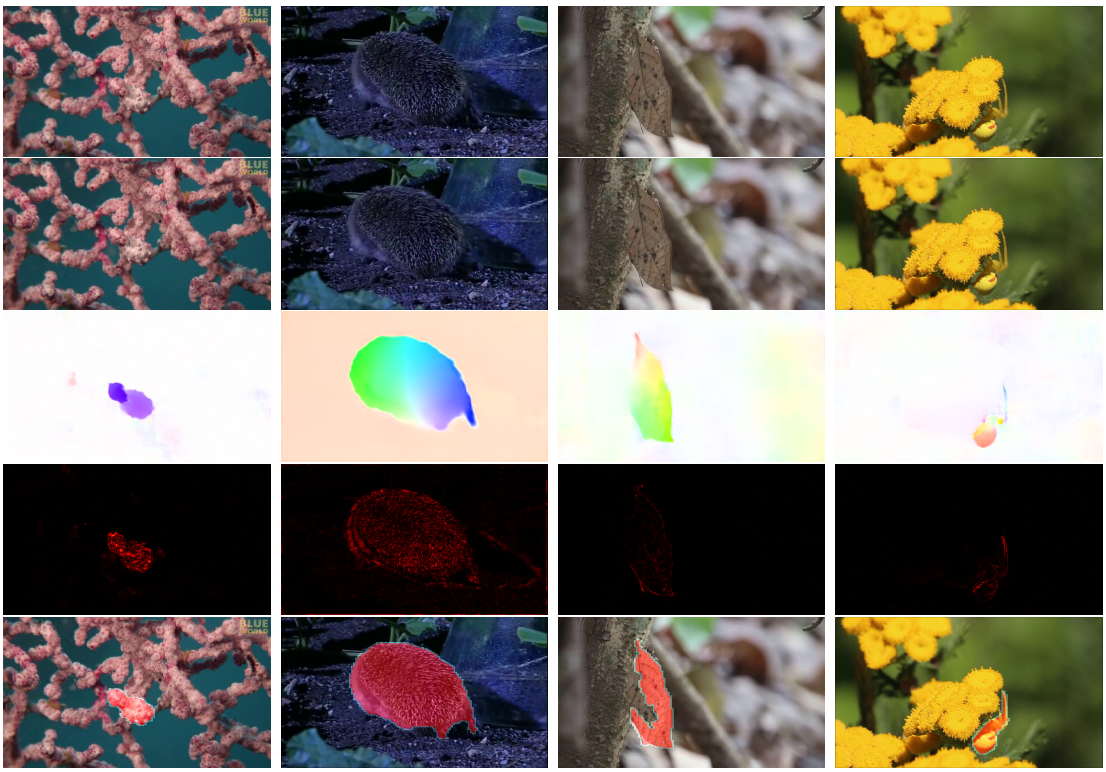


Figure 2.6: Motion segmentation results on MoCA. From top to bottom: frame t, frame $t+1$, PWCNet optical flow, aligned image difference, moving object segmentation.

show significantly stronger performance on DAVIS than on MoCA. Given this difference, our flow-based architecture still shows very competitive performance. Moreover, when we extend our model with an RGB appearance stream, we do observe a performance boost, but since our appearance model has only been fine-tuned on the DAVIS training set (30 training sequences), the two stream (ours-E) is not comparable with other models trained with more segmentation data.

Table 2.3: Results on the validation set of DAVIS 2016 of our method against MPNet[Tokmakov et al. 2017a], AD[Z. Yang et al. 2019], COSNet[Xiankai et al. 2019], LVO[Tokmakov et al. 2019], Seg-det[Dave et al. 2019] and Seg-track[Dave et al. 2019].

		Flow-based		RGB-based		Two-stream			
	Measure	MPNet	**ours-C**	AD	COSNet	LVO	Seg-det	Seg-track	**ours-E**
$\mathcal{J}$	Mean	60.3	**65.3**	**80.3**	77.6	69.8	**76.8**	75.8	69.9
	Recall	69.9	**77.3**	90.0	**91.4**	83.8	84.8	81.8	**85.3**
$\mathcal{F}$	Mean	58.7	**65.1**	**79.3**	77.5	70.1	**77.8**	76.1	70.3
	Recall	64.3	**74.7**	84.7	**87.4**	84.3	**91.3**	88.7	82.9

2.6 Conclusions

To summarise, in this paper we consider the problem of breaking animal camouflage in videos. Specifically, we propose a novel and effective architecture with two components: a differentiable registration module to highlight object boundaries; and a motion segmentation module with memory that discovers moving regions. As future work, we propose to improve the architecture from two aspects: *First*, building more effective memory module for handling longer video sequences, for example, a Transformer [Vaswani et al. 2017]. *Second*, for objects that are only partially moving, RGB appearance is required to get the sense of objectness, therefore a future direction is to explore RGB inputs, for both visual-matching-based registration and appearance features.

Acknowledgements

This research was supported by the UK EPSRC CDT in AIMS, Schlumberger Studentship, and the UK EPSRC Programme Grant Seebibyte EP/M013774/1.

Appendix

2.A Homography Transformation Estimation

This section provides further details on the estimation of the homography transformation described in 2.3.2. Consider the matrix A referenced in equation 2.4 as a standard Direct Linear Transformation (DLT) [Hartley and Zisserman 2004] mapping of a grid of source points to their corresponding target points, *i.e.* $\Omega = \{({x_i}^s, {y_i}^s, {x_i}^t, {y_i}^t), i \in [\![1, N]\!]\}$. Specifically, A is expressed as the following:

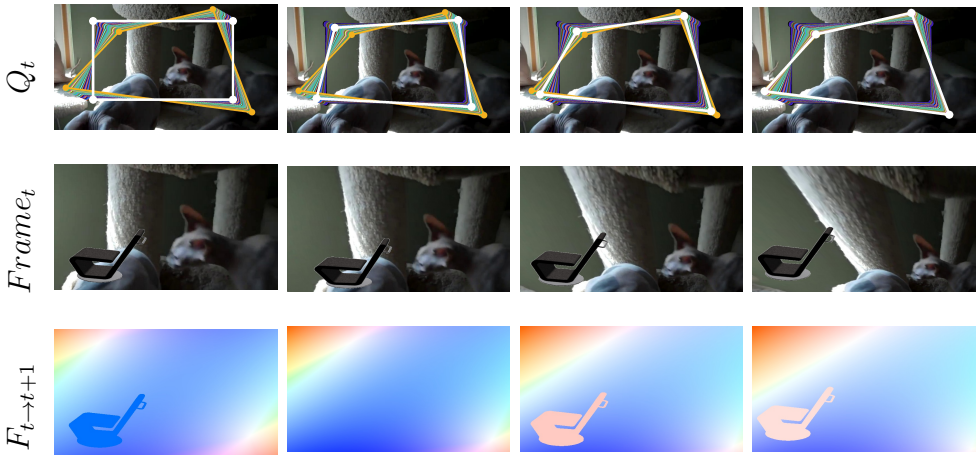
$$A = \begin{pmatrix} {x_1}^s & {y_1}^s & 1 & 0 & 0 & 0 & -{x_1}^s{x_1}^t & -{y_1}^s{x_1}^t & -{x_1}^t \\ 0 & 0 & 0 & {x_1}^s & {y_1}^s & 1 & -{x_1}^s{y_1}^t & -{y_1}^s{y_1}^t & -{y_1}^t \\ & & & & \vdots & & & & \\ {x_i}^s & {y_i}^s & 1 & 0 & 0 & 0 & -{x_i}^s{x_i}^t & -{y_i}^s{x_i}^t & -{x_i}^t \\ 0 & 0 & 0 & {x_i}^s & {y_i}^s & 1 & -{x_i}^s{y_i}^t & -{y_i}^s{y_i}^t & -{y_i}^t \\ & & & & \vdots & & & & \\ {x_N}^s & {y_N}^s & 1 & 0 & 0 & 0 & -{x_N}^s{x_N}^t & -{y_N}^s{x_N}^t & -{x_N}^t \\ 0 & 0 & 0 & {x_N}^s & {y_N}^s & 1 & -{x_N}^s{y_N}^t & -{y_N}^s{y_N}^t & -{y_N}^t \end{pmatrix}$$

Note that each pair of corresponding points constitutes two rows in the matrix A, and H has 8 degrees of freedom. Hence, traditional methods find a minimal set of 4 pairs of linearly independent corresponding points. We use a normalized equidistant grid Ω of $N = 64 \times 64$ correspondences derived from the forward optical flow mapping: $\Omega = \{({x_i}^s, {y_i}^s, {x_i}^s + {F^x}_{s\to t}({x_i}^s, {y_i}^s), {y_i}^s + {F^y}_{s\to t}({x_i}^s, {y_i}^s)), i \in [\![1, N]\!]\}$.

Finally, as in [Ranftl and Koltun 2018], the corresponding homography matrix is estimated via a singular value decomposition: $U\Sigma V^T = A^T diag(w) A$, with $h = \frac{v_0}{\|v_0\|}$ where v_0 is the right singular vector corresponding to the smallest eigen value.

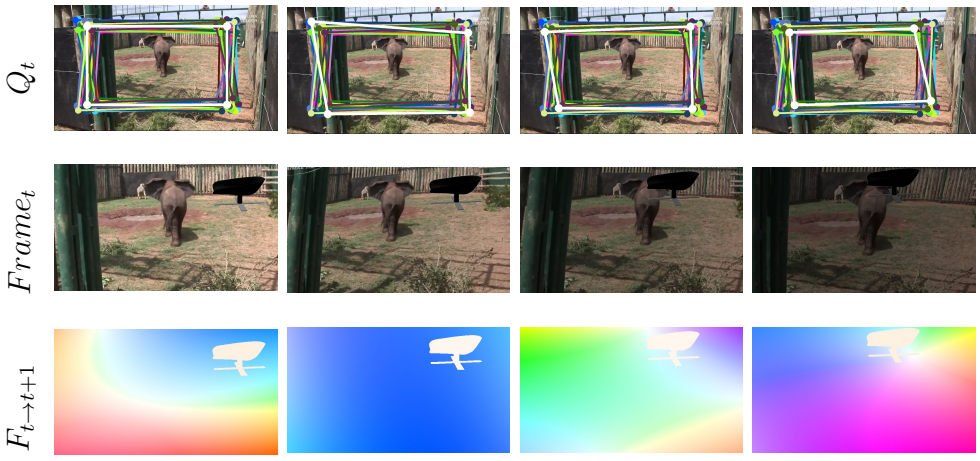
2.B Synthetic Moving Chairs

Our generation protocol is closely inspired by the method described in [DeTone et al. 2016], which we extend from a pair of images to a sequence. Namely, we generate a sequence from a single background image by applying a sequence of homographies. We first select the initial $F_{t=0}$ frame by cropping a random rectangle of which we jitter the vertices in order to generate the quadrilaterals defining the sequence images. For each quadrilateral $Q_{t=t_i} \neq Q_{t=0}$ we compute the underlying homography, *i.e.* mapping $Q_{t=t_i}$ to $Q_{t=t_0}$, and apply it to the image $F_{t=0}$ to obtain $F_{t=t_i}$. We consider two types of sequences: continuous sequences (Figure 2.B.1), computed via linear interpolation, to mimic a continuous camera motion; and random sequences (Figure 2.B.2), to simulate a shaking camera scenario. We further incorporate momentarily static objects and brightness change to imitate real cases.



(a) $t = 0$ (b) $t = 10$ (c) $t = 20$ (d) $t = 28$

Figure 2.B.1: Examples from a continuous sequence with a momentarily static object towards $t = 10$. The corresponding quadrilateral is represented in white



(a) $t = 0$ (b) $t = 10$ (c) $t = 20$ (d) $t = 28$

Figure 2.B.2: Examples from a random sequence with brightness change. The corresponding quadrilateral is represented in white

2.C End-to-end Motion Segmentation Pipeline

In this section, we provide a detailed description of the main modules of our end-to-end motion segmentation pipeline (Figure 2.2).

Differentiable registration.

We adopt the model introduced in [Moo Yi et al. 2018] totalling 12 residual blocks where each block is composed of two linear layers, separated by a global context normalization (GCN), batch normalization (BN1d) and ReLU (Table 2.C.1).

Architecture	Input $\in \mathcal{R}^{N\times4\times64\times64}$	Output
Registration	Reshape	$N \times 4 \times 4096$
	Conv1d, $(1, 4, 128)$	$N \times 128 \times 4096$
	Residual block: $\begin{bmatrix}\text{Conv1d}, (1, 128, 128)\\ \text{GCN}\\ \text{BN1d} + \text{ReLU}\\ \text{Conv1d}, (1, 128, 128)\\ \text{GCN}\\ \text{BN1d} + \text{ReLU}\end{bmatrix} \times 12$	$N \times 128 \times 4096$
	$\begin{bmatrix}\text{Conv1d}, (1, 128, 1)\\ \text{Sigmoid}\end{bmatrix}$	$N \times 1 \times 4096$

Table 2.C.1: Architecture of the differentiable registration module.

Note that the GCN, as defined in [Moo Yi et al. 2018], is a completely non-parametric instance normalization-like layer where each perceptron is normalized across the correspondences of each pair of images separately: $GCN(o^l{}_i) = \frac{o^l{}_i - \overline{o^l{}_i}}{std(o^l{}_i)}$ where $o^l{}_i \in \mathcal{R}^d$ denotes the output of the perceptron l for the correspondence i and $\overline{o^l{}_i}$ and $std(o^l{}_i)$ respectively the mean and standard deviation of the distribution of the output for all correspondences $\{o^l{}_i\}_{i\in[\![1,N]\!]}$.

Motion Segmentation Encoder.

Here, we adopt a variant of ResNet18 architecture described in Table 2.C.2.

Memory Module.

We adopt a similar architecture to [Tokmakov et al. 2019]. For a sequence with 11 frames, a convGRU cell of $hidden_size = 64$ and $kernel_size = 7$ is applied, forward and backward, to the $N \times 11 \times 256 \times 64 \times 64$ output of the encoder, where the second dimension refers to the sequence size. This results in $feat_fwd$ and $feat_bwd$ both of dimension $(N \times 11) \times 64 \times 64 \times 64$. These features are further concatenated along the channel dimension and fed to a conv2d of $kernel_size = 3, padding = 1$ to produce a bidirectional memory feature of size: $(N \times 11) \times 64 \times 64 \times 64$.

Motion Segmentation Decoder.

The decoder is composed of a residual block and the refinement block introduced in [Wug Oh et al. 2018], taking the output of memory module and the output of Conv_block_1 from the encoder, via skip connections, and upscaling by a factor of 2 resulting in the final motion prediction of size $(N \times 11) \times 1 \times 128 \times 128$.

	Input $\in \mathcal{R}^{N\times 256\times 256\times 3}$	Outputs
Encoder	Conv_block_1: $\begin{bmatrix}\text{Conv2d}, 7\times 7\, 3, 64, \text{stride}=2, \text{pad}=3\\ \text{BN2d}+\text{ReLU}\end{bmatrix}$	$N\times 64\times 128\times 128$
	Residual block_1: $\begin{bmatrix}\text{Conv2d}, 3\times 3, 64, 64, \text{stride}=2, \text{pad}=1\\ \text{BN2d}+\text{ReLU}\\ \text{Conv2d}, 3\times 3, 64, 64, \text{stride}=1, \text{pad}=1\\ \text{BN2d}+\text{ReLU}\\ \text{Conv2d}, 1\times 1, 64, 64, \text{stride}=2, \text{pad}=0\\ \text{BN2d}+\text{ReLU}\end{bmatrix}\times 2$	$N\times 128\times 64\times 64$
	Residual block_2: $\begin{bmatrix}\text{Conv2d}, 3\times 3, 64, 128, \text{stride}=3, \text{pad}=1\\ \text{BN2d}+\text{ReLU}\\ \text{Conv2d}, 3\times 3, 64, 128, \text{stride}=1, \text{pad}=1\\ \text{BN2d}+\text{ReLU}\\ \text{Conv2d}, 1\times 1, 64, 128, \text{stride}=3, \text{pad}=0\\ \text{BN2d}+\text{ReLU}\end{bmatrix}\times 2$	$N\times 128\times 64\times 64$
	Residual block_3: $\begin{bmatrix}\text{Conv2d}, 3\times 3, 128, 256, \text{stride}=2, \text{pad}=1\\ \text{BN2d}+\text{ReLU}\\ \text{Conv2d}, 3\times 3, 128, 256, \text{stride}=1, \text{pad}=1\\ \text{BN2d}+\text{ReLU}\\ \text{Conv2d}, 1\times 1, 128, 256, \text{stride}=2, \text{pad}=0\\ \text{BN2d}+\text{ReLU}\end{bmatrix}\times 4$	$N\times 256\times 64\times 64$

Table 2.C.2: Architecture of the motion encoder.

2.C.1 Qualitative results on MoCA

We provide further qualitative results of our model on the MoCA dataset in figure 2.C.1.

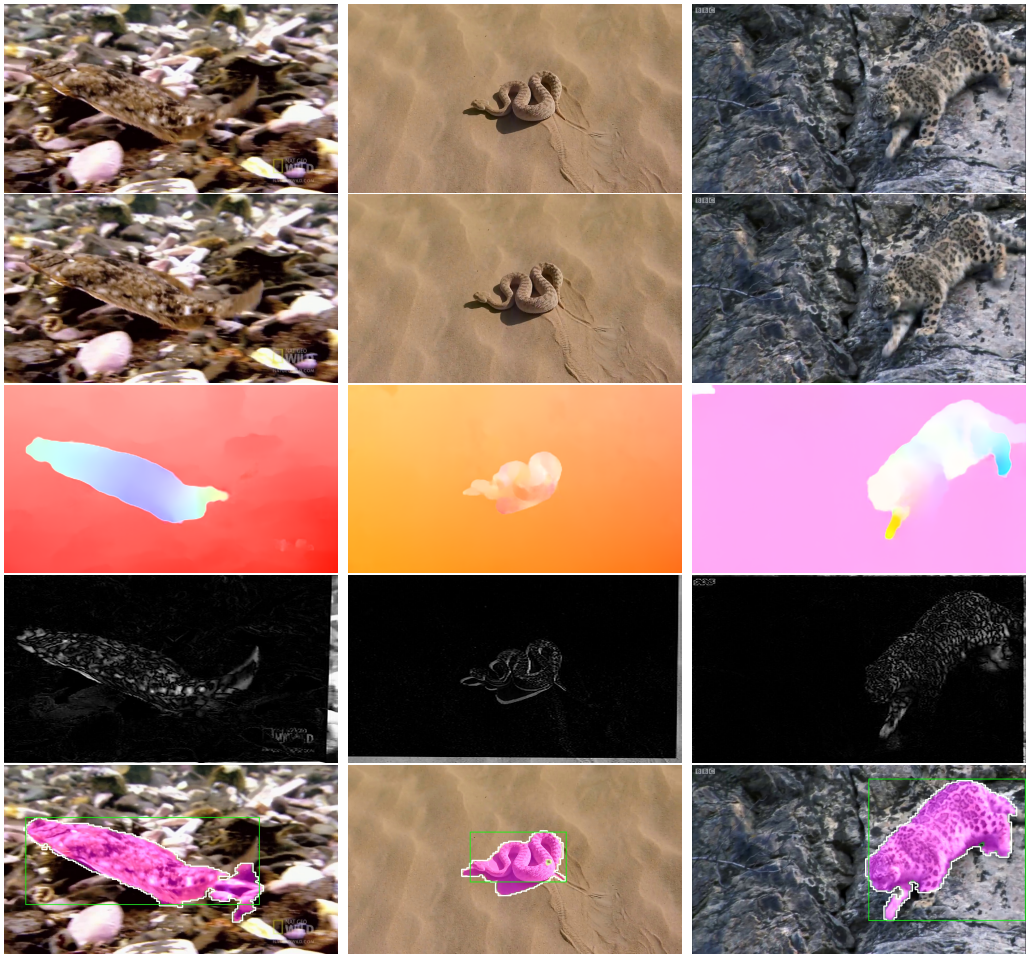


Figure 2.C.1: More motion segmentation results on MoCA. From top to bottom: frame t, frame $t+1$, PWCNet optical flow, aligned image difference, and the predicted moving object segmentation (the output of our model). We also show the ground truth annotation box in green

Chapter 3

Segmenting Invisible Moving Objects

This paper has been accepted for publication at the British Machine Vision Conference (BMVC), 2021.

Segmenting Invisible Moving Objects

Hala Lamdouar Weidi Xie Andrew Zisserman

Abstract

Biological visual systems are exceptionally good at perceiving objects that undergo changes in appearance, pose, and position. In this paper, we aim to train a computational model with similar functionality to segment the *moving* objects in videos. We target the challenging cases when objects are "invisible" in the RGB video sequence – for example, breaking camouflage, where visual appearance from a static scene can barely provide informative cues, or locating the objects as a whole even under partial occlusion.

To this end, we make the following contributions: (i) In order to train a motion segmentation model, we propose a scalable pipeline for generating synthetic training data, significantly reducing the requirements for labour-intensive annotations; (ii) We introduce a dual-head architecture (hybrid of ConvNets and Transformer) that takes a sequence of *optical flows* as input, and learns to segment the moving objects even when they are partially occluded or stop moving at certain points in videos; (iii) We conduct thorough ablation studies to analyse the critical components in data simulation, and validate the necessity of Transformer layers for aggregating temporal information and for developing object permanence. When evaluating on the MoCA camouflage dataset, the model trained only on synthetic data demonstrates state-of-the-art segmentation performance, even outperforming strong supervised approaches. In addition, we also evaluate on the popular benchmarks DAVIS2016 and SegTrackv2, and show competitive performance despite only processing optical flow. The project webpage is at: www.robots.ox.ac.uk/~vgg/research/simo/

3.1 Introduction

Object segmentation is undoubtedly one of the most widely researched problems in computer vision. Early attempts took inspiration from psychology, trying to understand the fundamental rules that cause people to see things as "unified" groups [Wertheimer 1923], and to translate them into the computational world, for instance, through clustering algorithms [J. Shi and Malik 2000] that grouped the pixels with certain characteristics, *e.g.* semantics, color, intensity, texture, and motion.

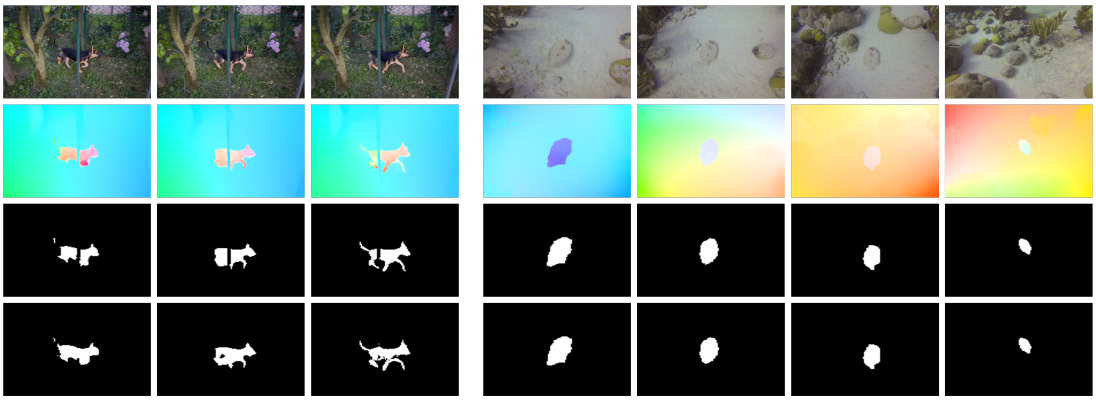


Figure 3.1.1: **Video sequences from DAVIS2016 (left) and MoCA (right).** Top to bottom: RGB sequence, optical flows, modal (only visible parts) and amodal (object as a whole) masks from our model prediction. Note that, our model exploits the optical flow as input, and is able to infer the whole mask of the objects even when they are partially occluded (the dog is partially occluded by the pole), or hidden in the environment (the fish is blended into the background environment).

In the recent literature, segmentation has been primarily treated as a problem of pixel-wise classification, and tackled by training deep networks on large-scale image or video datasets. This has two drawbacks: first, the requirement for heavy annotations limits the scalability; second, the assumption that objects can be well-identified via their visual appearances is often an over-simplification of our visual world, since objects may *not* be "visible" in a static scene. For instance, to segment objects that are camouflaged, or only partially visible due to occlusions (as shown in Figure 4.0.1), the visual appearance alone usually provides inadequate visual cues, thus challenging the existing segmentation models. Instead, motion cues are required for segmentation, and that is the objective of this paper.

To address the challenge of limited availability of groundtruth annotations, we propose a scalable pipeline for generating synthetic data, and advocate a *Sim2Real*

training procedure – once trained on synthetic data, the model can directly generalise to the downstream task, *e.g.* segmenting moving objects in real videos *without* finetuning on manual annotations. Generally speaking, although the visual scenes in real-world videos might be arbitrarily complex, the motion between two frames can always be factored into a combination of camera and object motion [Hartley and Zisserman 2004], effectively discarding the nuisance factors in visual appearance, *e.g.* color, texture, illumination. We exploit such observation to minimise the gap between the simulated and real videos. Specifically, for camera motion, it can often be approximated by a global transformation, while for an individual object, if it does move, pixels on it tend to form groups moving in similar direction and rate (and similarly in the optical flow). To this end, we simulate videos by composing moving objects onto a canvas that undergoes an independent transformation.

In practise, one challenge of only using optical flow comes from the fact that, objects in videos may stop moving or be partially occluded at any time point, leaving no effective cues for segmentation. As a consequence, one key ability of a functional vision system is to develop a sense of object permanence, *i.e.* realising that objects continue to exist even when they cannot be seen explicitly. To fulfil this goal, we propose a dual-head architecture, a hybrid of ConvNets and a Transformer, digesting a sequence of optical flows, and aggregating temporal information to flexibly output segmentation for only moving parts (modal), or the whole object even it is temporally static or under partial occlusion (amodal).

To summarise, we make the following contributions: (i) To address the challenge from limited availability of training data, we propose a scalable pipeline for generating synthetic video sequences with objects moving independently to the background motion; (ii) We introduce a dual-head architecture, a hybrid of ConvNet and Transformer, which takes a sequence of optical flow images as input, jointly processing them to segment the moving objects at any time point, and to predict both modal and amodal segmentations; (iii) After training on synthetic video sequences, we show *state-of-the-art* performance on the MoCA camouflage dataset, *without* even finetuning on manual annotations. In addition, we evaluate on two other popular benchmarks for video object segmentation, DAVIS2016 and SegTrackv2, and also show competitive performance.

3.2 Related Work

Video object segmentation considers the task of grouping pixels that belong to the same object. In the recent literature, semi-supervised video object segmentation (**semi-supervised VOS**), and unsupervised video object segmentation (**unsupervised VOS**) have attracted increasing attentions from the community [Bideau and Learned-Miller 2016a; N. Xu et al. 2018; Brox and Malik 2010; Ochs and Brox 2011; Papazoglou and Ferrari 2013; Jain et al. 2017; Tokmakov et al. 2019; Dave et al. 2019; Oh et al. 2019; Voigtlaender et al. 2019; Vondrick et al. 2018; J. Wang et al. 2019; Lai and W. Xie 2019; Lai et al. 2020; Maninis et al. 2018; Voigtlaender and Leibe 2017; Caelles et al. 2017; Fragkiadaki et al. 2012; Keuper et al. 2015; Jun Koh and C. Kim 2017; D. Fan et al. 2019; Z. Yang et al. 2019; Pont-Tuset et al. 2017; Jabri et al. 2020]. Specifically, semi-supervised VOS aims to continuously segment the objects that are specified by the user in the first frame; unsupervised VOS considers to localise and segment the *prominent* object in a video sequence. In practise, all previous approaches heavily rely on manual annotations, as an alternative, in this work, we consider the problem of video object segmentation from the perspective of *Sim2Real*, by that we mean to train the model on synthetic data, and demonstrate direct generalisation to real video sequences.

Motion segmentation shares some similarity with unsupervised VOS, but focuses only on the *moving* objects. In the literature, motion segmentation generally aims to segment the objects that move independently to the camera. Specifically, [Brox and Malik 2010; Ochs and Brox 2011; C. Xie et al. 2019] tackle the problem through the lens of clustering, with the goal to assign the same labels to pixels undergoing similar motion patterns; while [Tokmakov et al. 2017a; Tokmakov et al. 2019; Dave et al. 2019] train deep networks that map the motion fields to segmentation masks. In [Bideau and Learned-Miller 2016b; Bideau et al. 2018; Lamdouar et al. 2020; G. Yang and Ramanan 2018], the authors propose to highlight the independently moving object by compensating the background motion, either by registering consecutive frames, or explicitly estimating camera motions. [Y. Yang et al. 2019] proposes an adversarial setting, where a generator is trained to produce masks, jittering the input flow, such that the inpainter fails to estimate

the missing information. In constrained scenarios, such as the autonomous driving domain, [Ranjan et al. 2019] proposes to jointly optimise depth, camera motion, optical flow and motion segmentation. Concurrent work [C. Yang et al. 2021] adopts a layered representation to train a generative model, with motion segmentation being a by-product. In contrast to the existing approaches, we propose to train the architecture purely with synthetic video sequences, taking no prior knowledge of the objects' category or shape, and advocate direct generalisation to real videos.

Camouflage breaking aims to discover the objects that are hiding in the scene, where the visual appearance barely provides informative cues [Bideau and Learned-Miller 2016a; Le et al. 2016; Lamdouar et al. 2020]. As such, the objects will only be apparent when they start to move, thus is closely related to motion segmentation.

Optical flow usually treated as the default motion representation in computer vision applications, where synthetic data has been predominantly adopted for training models, e.g. MPI Sintel [Butler et al. 2012], FlyingThings3D [Mayer et al. 2016] and Monkaa [Mayer et al. 2016], pose estimation [Doersch and Zisserman 2019], AutoFlow [Sun et al. 2021].

Amodal segmentation refers to the task of segmenting the object as whole, including the portions that are partially occluded [Zhu et al. 2017; K. Li and Malik 2016; Purkait et al. 2019]. Part of this work follows this line of research and aims to extend amodal perception to video object segmentation.

3.3 Synthetic Data Generation

In this section, we describe a scalable pipeline for generating synthetic video sequences to train our proposed motion segmentation model. Specifically, we generate RGB sequences with objects being textured with samples from the DTD texture dataset [Cimpoi et al. 2014], the optical flow sequences can either be computed from groundtruth transformations, or by an off-the-shelf flow estimator, *e.g.* RAFT [Teed and Deng 2020]. In the following sections, we detail the generation process, and show an example of the generated video sequence in Figure 3.3.3.

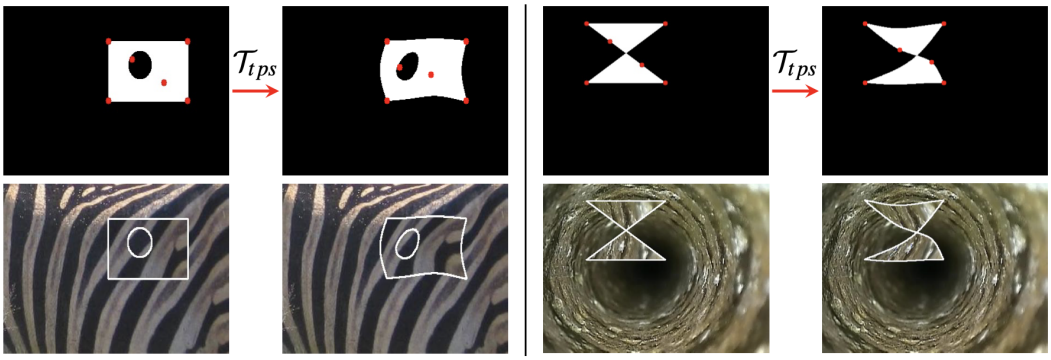
We refer the reader to Appendix 3.A for more figures and details on the generation procedure.

3.3.1 Foreground Objects

Synthetic shapes. We start with simple 2D shapes, namely a polygonal sprite, which is generated from 4-vertex polygons, convex and non-convex, with random holes. To increase the complexity, we occasionally replace polygons with real objects masks, for example, silhouettes that are sourced from a large scale dataset, e.g. YouTube-VOS.

Non-rigid objects. Apart from affine transformations on the entire object, we also generate non-rigid motions by applying thin plate splines $\mathcal{T}_{tps}$ with 6 control points. Let $P_{tl}, P_{tr}, P_{br}, P_{bl}$ denote the vertices of the generated polygon starting from top left to bottom left. The control points are chosen at these vertices and two additional points located at $(x_{P_{tl}} + 0.3 * w, y_{P_{tl}} + 0.3 * h)$ and $(x_{P_{br}} - 0.3 * w, y_{P_{br}} - 0.3 * h)$, where h and w stand for the maximal height and width of the polygon.

Articulated objects. To simulate the articulated objects, we split the object into 2 to 4 parts and apply different 2D rotations around the designated articulation vertices, with rotation angles $\theta \in \left[-\frac{\pi}{3}, +\frac{\pi}{3}\right]$.



(a) convex polygon with hole (b) non-convex polygon

Figure 3.3.1: **Generating foreground objects.** Generated object masks with thin plate spline $\mathcal{T}_{tps}$, the control points are shown in red.

3.3.2 Generating Motion Sequences

At this stage, we compose moving objects onto a canvas that undergoes an independent transformation, shown in Figure 3.3.2. For the optical flow, we consider

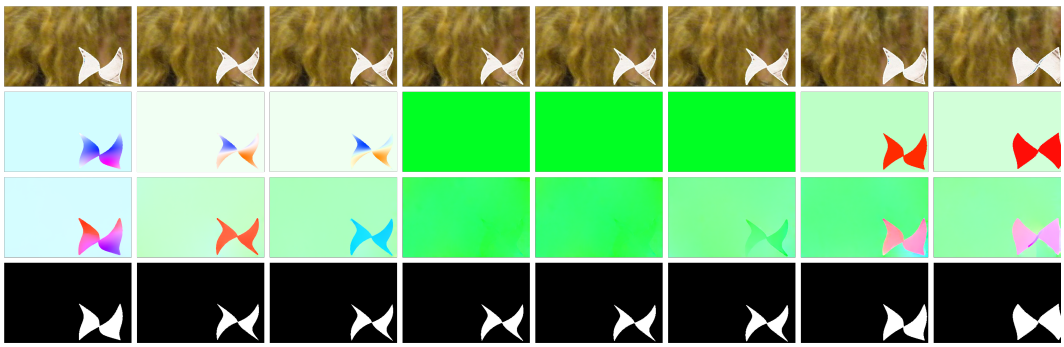


Figure 3.3.2: **Sequence of moving sprite undergoing a static subsequence.** From top to bottom: RGB frames, groundtruth optical flows, RAFT flows, segmentation masks.

two options: one is to use the groundtruth flow based on the applied transformations, and the other is to run an off-the-shelf flow estimator on the RGB sequence, *e.g.* RAFT. In the latter case, the background scenes can also be sourced from other videos. Additionally, we also incorporate static sub-sequences at various temporal locations, with the foreground object following the same transformation as the background. In such cases, objects will temporally disappear in the flow field, forcing the model to develop a sense of object permanence through temporal information.

3.3.3 Artificial Occluders

To simulate partial occlusions, we superimpose occluders on the canvas. They can be of arbitrary shapes, and follow the *same* motion as the background. As the occluders' trajectories intersect with the foreground object, the generated groundtruth segmentation masks will fall into two categories: masks that only delineate the visible parts of the object; or masks that delineate the objects as a whole, including the occluded parts. Both are shown in Figure 3.3.3.

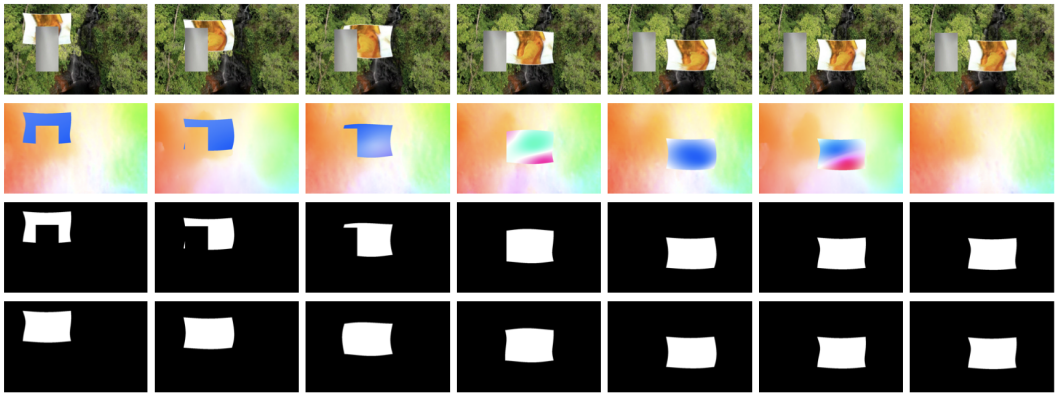


Figure 3.3.3: **Sequence of a moving sprite undergoing occlusion.** From top to bottom: RGB frames with the background sourced from a real video, and occluders are selected from different textures; optical flow sequence that is computed with RAFT; visible object (modal) segmentation; amodal segmentation.

3.4 Architecture

In this section, we present our proposed architecture for robustly segmenting moving objects, as illustrated in Figure 3.4.1. It is composed of a hybrid architecture, with ConvNets and Transformers. Specifically, the ConvNets are used to extract features from each input optical flow image, ending up as a sequence of feature maps, with the Transformer built on top to aggregate spatial-temporal information, and output framewise segmentations.

Motion Encoder. Given a sequence of optical flows computed from consecutive frames, $X = \{I_t, I_{t+1}, ...I_{t+k-1}\} \in \mathbb{R}^{k \times C_0 \times H_0 \times W_0}$, we process the motion fields with a ConvNet:

$$\{f_t, f_{t+1}, \ldots, f_{t+k-1}\} = \Phi_{\text{ENC}}(X)$$

where $f_i \in \mathbb{R}^{C \times H \times W}$ refers to the feature map, H, W, C denote height, width and channels.

Transformer Encoder. As inputs to the Transformer Encoder, we reshape the feature maps to a sequence of tokens:

$$\mathbb{V} = \textsc{Reshape}(\{f_t, f_{t+1}, \ldots, f_{t+k-1}\}) + \textsc{Pos}_t + \textsc{Pos}_s$$

with $\mathbb{V} \in \mathbb{R}^{C \times kHW}$ representing a total of kHW vectors. To keep track of the spatial-temporal position for each token, learnable spatial ($\textsc{Pos}_s$) and temporal ($\textsc{Pos}_t$) encodings are also added. To this end, we pass the sequence of vectors

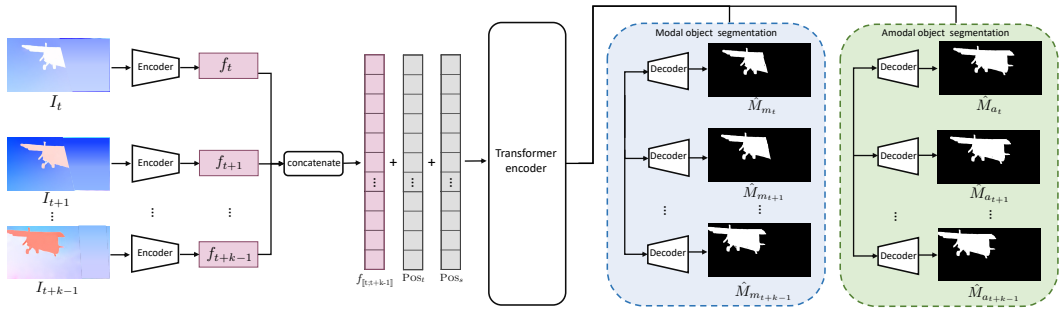


Figure 3.4.1: **Illustration of our proposed dual-head hybrid architecture for independent motion segmentation.** Our model takes a sequence of optical flows as input, and encodes each with a light-weight ConvNet, then positional spatio-temporal information is added to the outputted features $f_{[t,t+k-1]}$ before passing through the transformer encoder. Lastly, the output is fed into the dual-head decoder, predicting amodal $\hat{M}_{a_i}$ and modal masks $\hat{M}_{m_i}$.

into a standard Transformer Encoder with N layers ($N = 3$ in our case), each consisting of a stack of Multi-Head self-Attention, Layer Normalisation, and residual connections:

$$\overline{\mathbb{V}} = \textsc{Transformer-Encoder}(\mathbb{V})$$

$\overline{\mathbb{V}} \in \mathbb{R}^{C \times kHW}$ refers to the output from the Transformer Encoder.

Discussion. Unlike existing video segmentation works that mainly rely on RGB sequences, we only use optical flow as input. This has both pros and cons. On the one hand, flow has been shown to be highly effective in discarding the nuisance factors in visual appearance, *e.g.* texture, illumination, greatly facilitating the *Sim2Real* procedure to generalise towards real videos. On the other hand, it also brings challenges to segment the objects if they stop moving or are partially occluded at certain points of the video, as they cannot be seen explicitly in the flow fields. At this point, the Transformer layers play a critical role for aggregating temporal information, effectively capturing the object permanence, *i.e.* the object does not vanish even if it stops moving, instead, it just stays in the same place. In Section 4.5, we experimentally validate the usefulness of such Transformer layers.

Dual-Head Decoder. The outputs from the Transformer Encoder are passed into a dual-head decoder, with one head dedicated to predicting framewise amodal moving object masks $\hat{M}_{a_i}$, and the other one focuses on the visible or modal moving object masks $\hat{M}_{m_i}$, as shown in Figure 3.4.1. Symmetrically, the decoder uses a similar architecture to the encoder, with convolution transposes to recover

high resolution, and incorporating skip layers in a U-Net-like manner to fuse fine-grained features. We refer readers to Appendix 3.B for more detailed architecture descriptions.

Training. We train the hybrid dual-head architecture using pixel-wise classification:

$$\mathcal{L} = \frac{1}{k} \sum_{i=t}^{t+k-1} (\mathcal{L}_{BCE}(\hat{M}_{m_i}, M_{m_i}) + \mathcal{L}_{BCE}(\hat{M}_{a_i}, M_{a_i}))$$

3.5 Experiments

In this section, we start by describing the evaluation benchmarks and metrics, followed by the implementation details.

3.5.1 Evaluation Benchmarks

We conduct evaluation on four different datasets, three of them are used for unsupervised video object segmentation, and one synthetic dataset to test amodal segmentation.

DAVIS2016 [Perazzi et al. 2016] contains a total of 50 sequences (30 for training and 20 for validation), depicting diverse moving objects such as animals, people, and cars. The dataset contains 3455 1080p frames with pixel-wise annotations for the predominantly moving objects.

SegTrackv2 [F. Li et al. 2013] contains 14 sequences and 976 annotated frames. Each sequence has 1-6 moving objects, with challenges from motion blur, appearance change, complex deformation, occlusion, slow motion, and interacting objects.

Moving Camouflaged Animals (MoCA) [Lamdouar et al. 2020] contains 141 HD video sequences, depicting 67 kinds of camouflaged animals moving in natural scenes. Both temporal and spatial annotations are provided in the form of tight bounding boxes for every 5th frame. Following [C. Yang et al. 2021], we use the provided motion labels (locomotion, deformation, static) to filter out videos with predominantly no locomotion, resulting in 88 video sequences and 4803 frames.

Dataset for Amodal Segmentation (AMSeg). We evaluate on two datasets for amodal segmentation. Specifically, in terms of quantitative evaluation, we test on a synthetic dataset generated with moving objects and occluder, totalling 500 sequences with $15K$ frames. While for qualitative visualisation, we use a subset of DAVIS2016 sequences, with the masks manually completed for occluded regions. More sample sequences can be found in Appendix 3.A. We will release the generated sequences and annotations in the project webpage.

3.5.2 Evaluation Metrics

Depending on the provided annotations, we use two different evaluation metrics.

Segmentation (Jaccard). For DAVIS2016, SegTrackv2, and AMSeg, pixelwise segmentations are available, we report region similarity ($\mathcal{J}$) over the test set. For SegTrackv2, we follow the common practice [Y. Yang et al. 2019; Jain et al. 2017] and combine multiple objects as one single foreground.

Localization (Jaccard & Success Rate). For MoCA, as only bounding box annotations are provided, we report results in the form of detection success rate [Everingham et al. 2010; T. Lin et al. 2014], under different IoU thresholds ($\tau \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$).

3.5.3 Implementation Details

To get flow sequences on synthetic and real videos, we adopt off-the-shelf estimators, *e.g.* PWCNet [Sun et al. 2018], RAFT [Teed and Deng 2020]. For all datasets, optical flows are computed at the original resolution between image pairs, and converted into 3-channel images with the standard colorwheel used in the flow community [Baker et al. 2011; C. Yang et al. 2021]. This transformation is equivalent to the vector field representation with the advantage of offering an easier visualisation of the flow sequences. Our architecture takes as input sequence of size $X \in \mathbb{R}^{k \times 3 \times H_0 \times W_0}$, $H_0 = W_0 = 128$. Note that, for simplicity, all notations here have ignored the *batch* dimension (we use a batch size of 8 for training). The Motion Encoder projects the input samples to feature maps with smaller spatial

resolution, $\mathbb{R}^{k\times C\times H\times W}$, with $C = 512$, $H = \frac{H_0}{16}$ and $W = \frac{W_0}{16}$. For the Transformer Encoder, we use multi-head attention with 8 heads, and 3 layers. During training, we adopt the Adam optimiser with a learning rate of 5×10^{-4}. Instead of densely processing k consecutive frames, we proceed with a random selection of $l < k$ frames with corresponding absolute positional encoding (within the original sequence of length k). Effectively, this operation acts as a temporal dropout of input flows, reducing the required memory for training long video sequences, here, we use $k = 16$ and $l = 8$ in our training. At inference time, we process the entire sequence with all the k temporal samples.

3.6 Results

Here, we conduct ablation studies by varying one variable at a time, *e.g.* critical components for data simulation, and necessity of transformer for developing object permanence. After that, we compare with other state-of-the-art approaches on different benchmarks.

3.6.1 Ablation Studies

As shown in Table 3.6.1, we report the results by training models on synthetic data generated with increasing complexities. Note that, all models are directly applied to MoCA, DAVIS2016 and AMSeg datasets, *without* any fine-tuning included.

Choice of optical flow. While comparing the results of models trained on different optical flows, two phenomenon can be observed: *First*, training on RAFT flows are preferable than on groundtruth flow, *e.g.* Ours-(B,C) vs. Ours-(D,E). As during inference time, RAFT flow is also used, effectively introducing minimal domain gaps between training and testing scenarios. *Second*, RAFT provides higher quality flow estimations compared to other methods, *e.g.* PWCNet, showing superior results *cf.* Ours-(F,G).

Choice of background motion. Unsurprisingly, while training our proposed model, simulating backgrounds with a homography transformation substantially outperforms that of only a static background, *e.g.* Ours-A vs. Ours-B; using frames from other videos as background, *i.e.* real motions are introduced, can further boost

	Sequence	Bg Motion	Flow	Occluder	Transformer	MoCA	DAVIS2016	AMSeg
Ours-A	OF	S	GT	✗	✓	23.1	10.4	11.4
Ours-B	OF	H	GT	✗	✓	54.1	44.2	71.5
Ours-C	OF	H	GT	✓	✓	52.5	48.8	69.2
Ours-D	RGB	H	RAFT	✗	✓	57.7	50.1	84.7
Ours-E	RGB	H	RAFT	✓	✓	59.9	52.2	90.3
Ours-F	**RGB**	**H + R**	**RAFT**	✓	✓	**68.6**	**67.8**	**91.0**
Ours-G	RGB	H + R	PWCNet	✓	✓	62.9	56.4	81.1
Ours-H	RGB	H + R	RAFT	✓	✗	61.3	65.0	84.4

Table 3.6.1: **Ablation study on synthetic data generation.** Performance is measured by Jaccard on all datasets. Specifically, **Sequence** denotes the format of simulated video sequence, in RGB or optical flow (OF) (note that, in all cases, our pipeline only takes as input the optical flow sequence, either created during the simulation process or computed from the generated RGB sequences); **Bg Motion** refers to the background motion used during data simulation (S: Static, H: Homography, R: Real motion); **Flow** refers to the sources of optical flows for training. During inference, all the optical flows are computed from RAFT.

the performance on downstream benchmarks, *e.g.* Ours-E vs. Ours-F.

Artificial occluder. While comparing Ours-E to Ours-D, we validate the usefulness of introducing artificial occluders for data simulation, this is especially true on AMSeg dataset, which is specifically designed to evaluate model's ability on amodal segmentation.

Transformer Encoder. We conduct experiment by ablating the Transformer Encoder, as shown by Ours-F vs. Ours-H, the former shows superior performance on all the datasets, clearly demonstrating the necessity of Transformer Encoder, potentially on cases where objects stop moving or partially occluded at certain time point.

3.6.2 Comparison with State-of-the-art

In this section, we take the best setting discovered from the ablation study (**Ours-F**), and compare its segmentation results with previous approaches on the public benchmarks, *e.g.* MoCA, DAVIS2016, SegTrackv2. Again, in all cases, we train the model on synthetic data, and directly evaluate on the downstream tasks, *without* using any manual annotations.

On MoCA

As shown in Table 5.6.4, we achieve the state-of-the-art results on MoCA, significantly outperforming the previous approach (CIS [Y. Yang et al. 2019] by over 19%, MoSeg [C. Yang et al. 2021] by over 4%). Notably, our proposed model even shows superior performance than the top supervised approaches, *e.g.* COSNet and MATNet.

					Success Rate					
Model	Sup.	RGB	Flow	$\mathcal{J} \uparrow$	$\tau = 0.5$	$\tau = 0.6$	$\tau = 0.7$	$\tau = 0.8$	$\tau = 0.9$	SR_{mean}
CIS	✗	✓	✓	49.4	0.556	0.463	0.329	0.176	0.030	0.311
MoSeg	✗	✗	✓	63.4	0.742	0.654	0.524	0.351	0.147	0.484
Ours	✗	✗	✓	**68.6**	**0.772**	**0.717**	**0.623**	0.464	**0.255**	**0.566**
COD	✓	✗	✓	44.9	0.414	0.330	0.235	0.140	0.059	0.236
COSNet	✓	✓	✗	50.7	0.588	0.534	0.457	0.337	0.167	0.417
MATNet	✓	✓	✓	64.2	0.712	0.670	0.599	**0.492**	0.246	0.544

Table 3.6.2: **Results on MoCA dataset.** Successful localization rate for various thresholds τ following the same metric proposed in [C. Yang et al. 2021]. We compare our method against CIS [Y. Yang et al. 2019], MoSeg [C. Yang et al. 2021], COD [Lamdouar et al. 2020], COSNet [Xiankai et al. 2019], MATNet [Tianfei Zhou et al. 2020]

On DAVIS2016 & SegTrackv2

In this section, we report results on DAVIS2016 and SegTrackv2. Notably, in these two benchmarks, motion is not playing the dominant role as it is in MoCA, as the objects can often be well-identified by their appearance. This can be observed from the results of COSNet, which shows significantly stronger performance on DAVIS than on MoCA, despite not using any motion information.

As shown in Table 3.6.3, our model trained on synthetic data still shows competitive performance, achieving state-of-the-art results on DAVIS2016 among all self-supervised learning approaches. While for SegTrackv2, the videos occasionally include multiple objects, with only a subset of them moving, in such case, no informative cues will support object segmentation in the flow input. We achieve competitive performance nonetheless.

Qualitative results. As shown in Figure 3.6.1, the following phenomenon can be observed: *First*, our model provides high-quality segmentation masks, despite



Figure 3.6.1: **Qualitative Results on DAVIS2016, MoCA, SegTrackv2.**

Model	Sup.	RGB	Flow	DAVIS16	STv2
SAGE [W. Wang et al. 2018]	✗	✓	✓	42.6	57.6
NLC [Faktor and Irani 2014]	✗	✓	✓	55.1	**67.2**
CUT [Keuper et al. 2015]	✗	✓	✓	55.2	54.3
FTS [Papazoglou and Ferrari 2013]	✗	✓	✓	55.8	47.8
CIS [Y. Yang et al. 2019]	✗	✓	✓	59.2	45.6
Ours	✗	✗	✓	**67.8**	62.0
SFL [J. Cheng et al. 2017]	✓	✓	✓	67.4	–
FSEG [Jain et al. 2017]	✓	✓	✓	70.7	61.4
LVO [Tokmakov et al. 2019]	✓	✓	✓	75.9	57.3
ARP [Song et al. 2018]	✓	✓	✓	76.2	57.2
COSNet [Xiankai et al. 2019]	✓	✓	✗	80.5	–
MATNet [Tianfei Zhou et al. 2020]	✓	✓	✓	82.4	–
3DC-Seg [S. Mahadevan et al. 2020]	✓	✓	✓	**84.3**	–

Table 3.6.3: **Results on DAVIS2016 and SegTrackv2.**

only being trained on optical flow computed from synthetic data; *Second*, the model is able to maintain the object permanence through frames. This is particularly evident for the second and fourth examples, where the flow fields fail to capture part of the moving object, or completely fail at a specific time step. Our model is able to recover the missing parts of the dancer in the second example, and the bird in the fourth column, while MATNet [Tianfei Zhou et al. 2020] only provides partial segmentations; *Third*, as can be seen in the third column, our amodal head is capable of estimating the correct shape of the object despite the occlusion by jumping poles.

3.7 Conclusions

To summarise, this paper considers the problem of segmenting *moving* objects in videos. Specifically, we propose a scalable pipeline for generating synthetic video sequences, with objects moving independently to the background motion. This has shown to be effective for resolving the challenge from limited availability of training data; In addition, we introduce a dual-head architecture (hybrid of ConvNet and Transformer), which enables to jointly process a sequence of optical flows, and segment the moving objects at any time point, outputting both modal (only visible parts) and amodal (object as a whole) segmentations. After training our proposed model on synthetic video sequences, we show *state-of-the-art* performance on the MoCA camouflage datasets, even outperforming supervised approaches. On DAVIS2016 and SegTrackv2, our proposed model also demonstrates competitive performance, despite not using *any* manual annotation.

Acknowledgements.

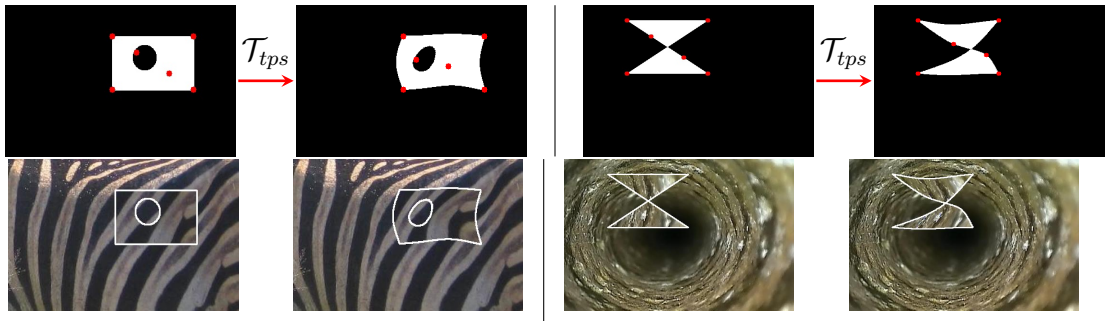
This research is supported by the UK EPSRC funded CDT in Autonomous Intelligent Machines and Systems (AIMS), the EPSRC Programme Grant VisualAI EP/T028572/1, a Schlumberger Studentship, and a Royal Society Research Professorship.

Appendix

3.A Synthetic Motion Simulation

3.A.1 Process of simulation of non-rigid objects

Figure 3.A.1 shows an example. Specifically, we apply a thin plate spline transform $\mathcal{T}_{tps}$ with 6 control points. Let $P_{tl}, P_{tr}, P_{br}, P_{bl}$ denote the vertices of the generated polygon starting from top left to bottom left. The control points are chosen at these vertices and two additional points located at $(x_{P_{tl}} + 0.3 * w, y_{P_{tl}} + 0.3 * h)$ and $(x_{P_{br}} - 0.3 * w, y_{P_{br}} - 0.3 * h)$, where h and w stand for the maximal height and width of the polygon. Note that for each sequence, we apply framewise random shifts to the control points.



(a) convex polygon with hole (b) non-convex polygon

Figure 3.A.1: **Example of deformed sprite objects generation.** Generated object masks with the control points used for the thin plate spline $\mathcal{T}_{tps}$ shown in red.

3.A.2 Sequence with static motion

Figure 3.A.2 shows an example sequence where the moving object is a deforming non-convex polygon that becomes static with respect to its background on the last two frames. Note that in this example, there is no occluder, hence the modal and

amodal segmentation masks are identical.

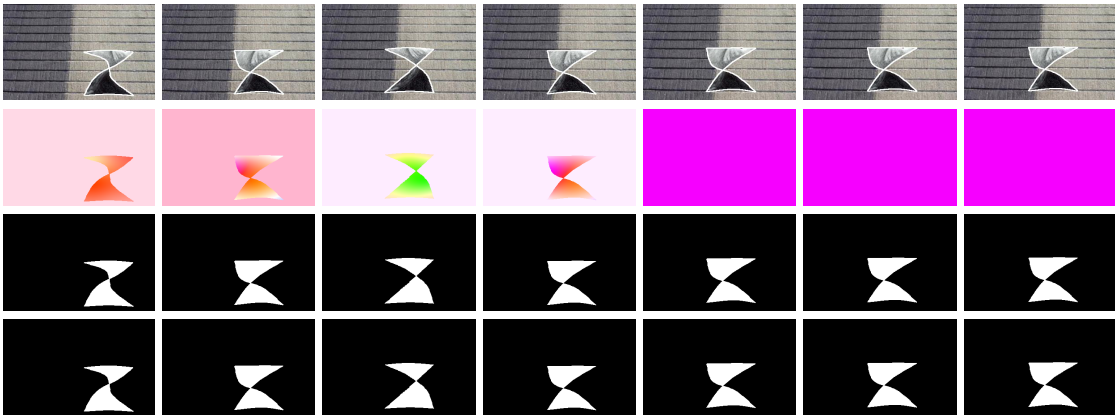


Figure 3.A.2: **Example sequence where the moving object becomes static.** *From top to bottom*: RGB sequence, optical flow, modal segmentations, amodal segmentations.

3.A.3 Moving objects passing behind an occluder

Figure 3.A.3 shows an example sequence where the moving sprite object passes behind an occluder which results in different modal and amodal segmentation masks.

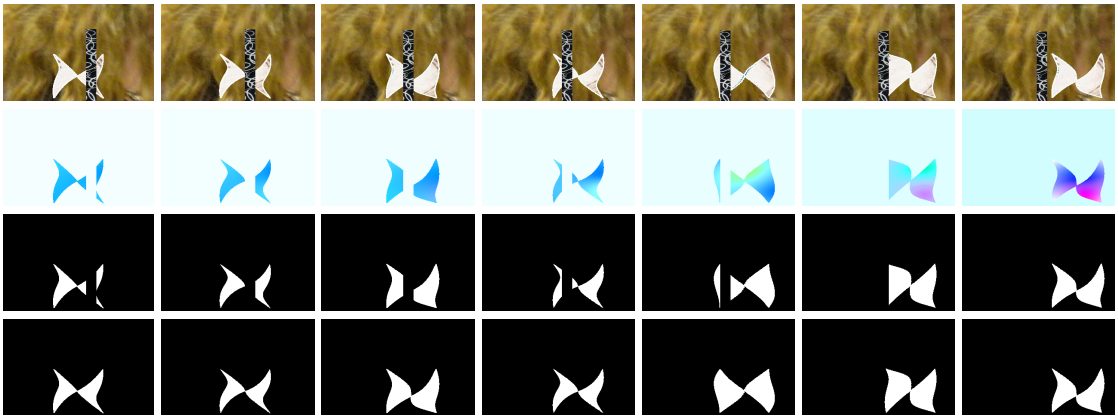


Figure 3.A.3: **Example sequence with objects passing behind an occluder.** *From top to bottom*: RGB sequence, optical flow, modal segmentations, amodal segmentations.

3.A.4 Real background motion.

Figure 3.A.4 shows an example of deforming convex polygon undergoing a translational motion, with the presence of a rectangular occluder. Unlike Figures 3.A.2 and 3.A.3, where the background motion is synthetic, *i.e.* a sequence of translations, Figure 3.A.4 presents a real background motion and the optical flow is

computed using RAFT [Teed and Deng 2020].

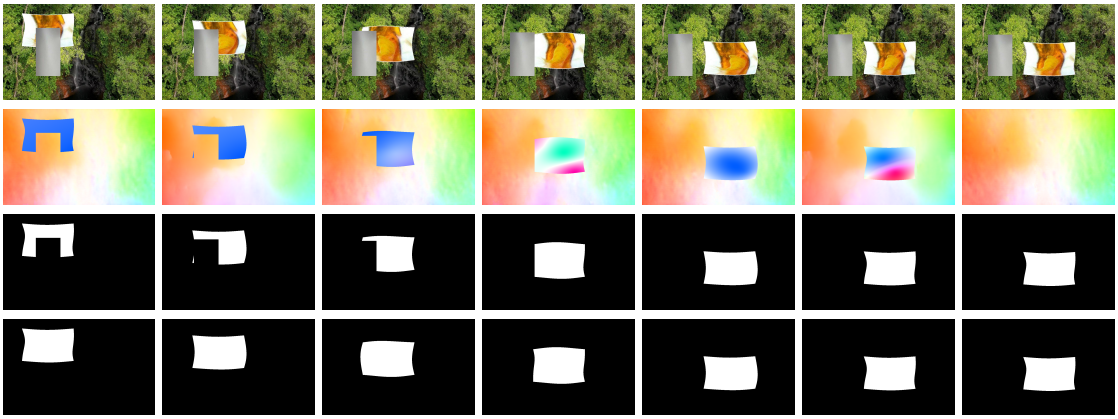


Figure 3.A.4: **Example sequence where the background sequence is taken from another video.** *From top to bottom*: RGB sequence, optical flow, modal segmentations, amodal segmentations.

3.A.5 Real shape and background motion.

Figure 3.A.5 presents a moving object (homographic transformation) under real background motion, in contrast to the previous step, now the shape can be an arbitrary silhouette. Note that we use different occluder shapes, e.g. rectangular in Figure 3.A.3 and Figure 3.A.4, and circular in Figure 3.A.5.

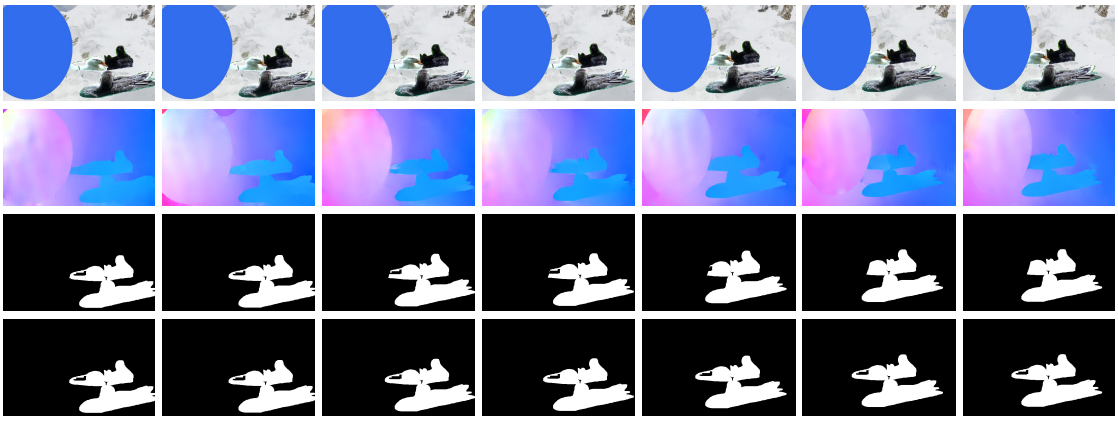


Figure 3.A.5: **Example sequence with homographic silhouette and background motion.** *From top to bottom*: RGB sequence, optical flow, modal segmentations, amodal segmentations.

3.B Dual-head Motion Segmentation Model

We present in Table 3.B.1 a detailed description of our overall architecture.

	stage	operation	output sizes
	input	–	$3 \times 128 \times 128$
Encoder	conv1	$[3 \times 3, 64] \times 2$	$64 \times 128 \times 128$
	mp1	maxpool, stride $= 2$	$64 \times 64 \times 64$
	conv2	$[3 \times 3, 128] \times 2$	$128 \times 64 \times 64$
	mp2	maxpool, stride $= 2$	$128 \times 32 \times 32$
	conv3	$[3 \times 3, 256] \times 2$	$256 \times 32 \times 32$
	mp3	maxpool, stride $= 2$	$256 \times 16 \times 16$
	conv4	$[3 \times 3, 512] \times 2$	$512 \times 16 \times 16$
	mp4	maxpool, stride $= 2$	$512 \times 8 \times 8$
	conv5	$[3 \times 3, 512] \times 2$	$512 \times 8 \times 8$
Transformer	t_{pos}	Embedding $[8 \times 512]$	$512 \times 8 \times 8$
	x_{pos}	Embedding $[8 \times 512]$	$512 \times 8 \times 8$
	y_{pos}	Embedding $[8 \times 512]$	$512 \times 8 \times 8$
	transEnc	input $= 512$, $n_{layers} = 3$ $n_{heads} = 8$, feedforward $= 1024$	512×512
Decoder	conv1	$[3 \times 3, 512] \times 2$	$512 \times 8 \times 8$
	conv^T1	$3 \times 3, 256, \text{stride} = 2$	$256 \times 16 \times 16$
	conv2	$[3 \times 3, 256] \times 2$	$256 \times 16 \times 16$
	conv^T2	$3 \times 3, 128, \text{stride} = 2$	$128 \times 32 \times 32$
	conv3	$[3 \times 3, 128] \times 2$	$128 \times 32 \times 32$
	conv^T3	$3 \times 3, 64, \text{stride} = 2$	$64 \times 64 \times 64$
	conv4	$[3 \times 3, 64] \times 2$	$64 \times 64 \times 64$
	conv^T4	$3 \times 3, 64, \text{stride} = 2$	$64 \times 128 \times 128$
	outconv	$7 \times 7, 64, \text{padding} = 3,$ $7 \times 7, 1$	$1 \times 128 \times 128$

Table 3.B.1: **Network architecture.** In this work, we adopt a ConvNet as visual encoder, and use Transformer Encoder for aggregating the temporal information, which is followed by ConvNet-based decoder to recover the resolution. Unless specified otherwise, all convolution layers have stride $= 1$ and padding $= 1$.

3.C Further Results

We present in Figure 3.C.1 examples of modal and amodal segmentation results from MoCA and DAVIS 2016 datasets and show sequences from SegTrackV2 in Figure 3.C.2.

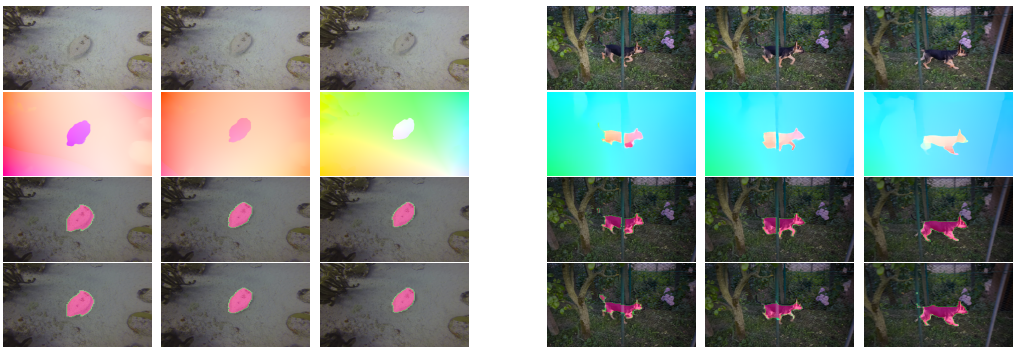


Figure 3.C.1: **Qualitative examples from MoCA and DAVIS2016**. *From top to bottom*: RGB sequence, optical flow, modal segmentations, amodal segmentations. Note that our amodal head is able to overcome degraded optical flow quality and provide a more complete segmentation in the right example.



Figure 3.C.2: **Qualitative examples from SegTrackv2 dataset**. *From top to bottom*: RGB sequence, optical flow, modal segmentations, moving object segmentations. Here modal and amodal predictions are identical.

Chapter 4

Self-supervised Video Object Segmentation by Motion Grouping

This paper has been accepted for publication at the International Conference on Computer Vision (ICCV), 2021.

Self-supervised Video Object Segmentation by Motion Grouping

Charig Yang Hala Lamdouar Erika Lu

Andrew Zisserman Weidi Xie

Visual Geometry Group, University of Oxford

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

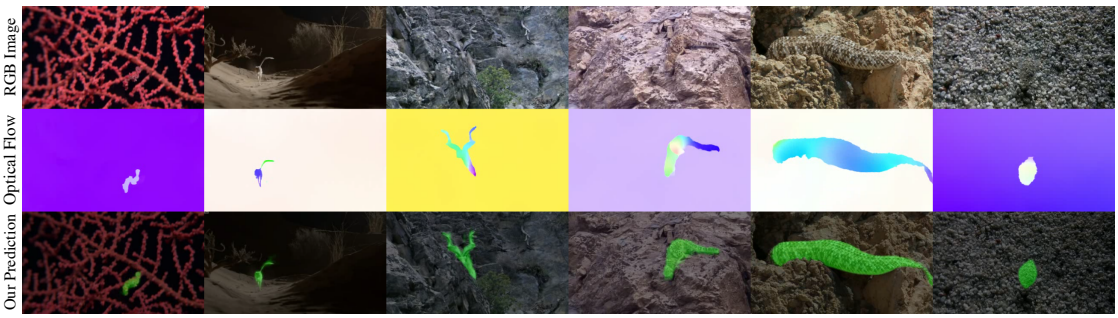


Figure 4.0.1: **Segmenting camouflaged animals.** Motion plays a critical role in augmenting the capability of our visual system for perceptual grouping in complex scenes – for example, in these sequences (MoCA dataset [Lamdouar et al. 2020]), the visual appearance (RGB images) is clearly uninformative. We propose a self-supervised approach to segment objects using *only* motion, i.e. optical flow. From top to bottom rows, we show the video frames, optical flow between consecutive frames, and the segmentation produced by our approach.

Abstract

Animals have evolved highly functional visual systems to understand motion, assisting perception even under complex environments. In this paper, we work towards developing a computer vision system able to segment objects by exploiting motion cues, i.e. motion segmentation. We make the following contributions: First, we introduce a simple variant of the Transformer to segment optical flow frames into primary objects and the background. Second, we train the architecture in a self-supervised manner, i.e. without using any manual annotations. Third, we analyze several critical components of our method and conduct thorough ablation studies to vali-

date their necessity. Fourth, we evaluate the proposed architecture on public benchmarks (DAVIS2016, SegTrackv2, and FBMS59). Despite using only optical flow as input, our approach achieves superior or comparable results to previous state-of-the-art self-supervised methods, while being an order of magnitude faster. We additionally evaluate on a challenging camouflage dataset (MoCA), significantly outperforming the other self-supervised approaches, and comparing favourably to the top supervised approach, highlighting the importance of motion cues, and the potential bias towards visual appearance in existing video segmentation models.

4.1 Introduction

When looking around the world, we effortlessly perceive a complex scene as a set of distinct objects. This phenomenon is referred to as *perceptual grouping* – the process of organizing the incoming visual information – and is usually considered a fundamental cognitive ability that enables understanding and interacting with the world efficiently. How do we accomplish such a remarkable perceptual achievement, given that the visual input is, in a sense, just a spatial distribution of variously colored individual points/pixels? In 1923, Wertheimer [Wertheimer 1923] first introduced the *Gestalt principles* with the goal of formulating the underlying causes by which sensory data is organized into groups, or Gestalten. The principles are much like heuristics with "a bag of tricks" [Ramachandran 1985] that the visual system may exploit for grouping, for example, proximity, similarity, closure, continuation, common fate, *etc.*

In computer vision, *perceptual grouping* is often closely related to the problem of segmentation, i.e. extracting the objects with arbitrary shape (pixel-wise labels) from cluttered scenes. In the recent literature of semantic or instance segmentation, tremendous progress has been made by training deep neural networks on image or video datasets. While it is exciting to see machines with the ability to detect, segment, and classify objects in images or video frames, training such segmentation models through supervised learning requires massive human annotation, and consequently limits their scalability. Even more importantly, the assumption that objects can be well-identified by their appearance alone in static frames is often an oversimplification – objects are not always visually distinguishable from

their background environment. For instance when trying to discover camouflaged animals/objects from the background (Figure 4.0.1), extra cues, such as motion or sound, are usually required.

Among the numerous cues, motion is usually simple to obtain as it can be implicitly generated from unlabeled videos. In this paper, we aim to exploit such cues for object segmentation in a self-supervised manner, i.e. *zero* human annotation is required for training. At a high level, we aim to exploit the common fate principle, with the basic assumption being that **elements tend to be perceived as a group if they move in the same direction at the same rate (have similar optical flow)**. Specifically, we tackle the problem by training a generative model that decomposes the optical flow into foreground (object) and background layers, describing each as a homogeneous field, with discontinuities occurring only between layers. We adopt a variant of the Transformer [Vaswani et al. 2017], with the self-attention being replaced by slot attention [Locatello et al. 2020], where iterative grouping and binding have been built into the architecture. With some critical architectural changes, we show that pixels undergoing similar motion are grouped together and assigned to the same layer.

To summarize, we make the following contributions: *first*, we introduce a simple architecture for video object segmentation by exploiting motions, using only optical flow as input. *Second*, we propose a self-supervised proxy task that is used to train the architecture without any manual supervision. *Third*, we conduct thorough ablation studies on the components that are key to the success of our architecture, such as a consistency loss on optical flow computed from various frame gaps. *Fourth*, we evaluate the proposed architecture on public benchmarks (DAVIS2016 [Perazzi et al. 2016], SegTrackv2 [F. Li et al. 2013], and FBMS59 [Peter et al. 2014]), outperforming previous state-of-the-art self-supervised models, with comparable performance to the supervised approaches. Moreover, we also evaluate on a camouflage dataset (MoCA [Lamdouar et al. 2020]), demonstrating a significant performance improvement over the other self- and supervised approaches, highlighting the importance of motion cues, and the potential bias towards visual appearance in existing video segmentation models.

4.2 Related Work

Video object segmentation has been a longstanding task in computer vision, which involves segmenting the pixels (or edges) belonging to an image into groups (e.g. objects). In recent literature [Bideau and Learned-Miller 2016a; Pont-Tuset et al. 2017; N. Xu et al. 2018; Brox and Malik 2010; Ochs and Brox 2011; Papazoglou and Ferrari 2013; Jain et al. 2017; Tokmakov et al. 2019; Dave et al. 2019; Oh et al. 2019; Voigtlaender et al. 2019; Vondrick et al. 2018; J. Wang et al. 2019; Lai and W. Xie 2019; Lai et al. 2020; Maninis et al. 2018; Voigtlaender and Leibe 2017; Caelles et al. 2017; Fragkiadaki et al. 2012; Keuper et al. 2015; Jun Koh and C. Kim 2017; D. Fan et al. 2019; Z. Yang et al. 2019], two protocols have attracted increasing interest from the vision community, namely, semi-supervised video object segmentation (**semi-supervised VOS**), and unsupervised video object segmentation (**unsupervised VOS**). The former aims to re-localize one or multiple targets that are specified in the first frame of a video with pixel-wise masks, and the latter considers automatically separating the object of interest (usually the most salient one) from the background in a video sequence. Despite being called **unsupervised VOS**, in practice, the popular methods to address such problems extensively rely on supervised training, for example, by using two-stream networks [Papazoglou and Ferrari 2013; Jain et al. 2017; Tokmakov et al. 2019; Dave et al. 2019] trained on large-scale external datasets. As an alternative, in this work, we consider a *completely unsupervised* problem, where no manual annotation is used for training whatsoever.

Motion segmentation shares some similarity with unsupervised VOS, but focuses on discovering *moving* objects. In the literature, [Brox and Malik 2010; Ochs and Brox 2011; Keuper et al. 2015; Sivic et al. 2004; C. Xie et al. 2019] consider clustering the pixels with similar motion patterns; [Tokmakov et al. 2017a; Tokmakov et al. 2019; Dave et al. 2019] train deep networks to map the motions to segmentation masks. Another line of work has tackled the problem by explicitly leveraging the independence, in the flow field, between the moving object and its background. For instance, [Y. Yang et al. 2019] proposes an adversarial setting, where a generator is trained to produce masks, altering the input flow, such that the inpainter fails to estimate the missing information. In [Bideau and Learned-

Miller 2016b; Bideau et al. 2018; Lamdouar et al. 2020], the authors propose to highlight the independently moving object by compensating for the background motion, either by registering consecutive frames, or explicitly estimating camera motion. In constrained scenarios, such as the autonomous driving domain, [Ranjan et al. 2019] proposes to jointly optimize depth, camera motion, optical flow and motion segmentation.

Optical flow computation is one of the fundamental tasks in computer vision. Deep learning methods allow efficient computation of optical flow, both in supervised learning on synthetic data [Sun et al. 2018; Teed and Deng 2020], and in the self-supervised [L. Liu et al. 2020; P. Liu et al. 2019] setting. Flow has been useful for a wide range of problems, occasionally even used in lieu of appearance cues (RGB images) for tracking [Sidenbladh et al. 2000], pose estimation [Doersch and Zisserman 2019], representation learning [Mahendran et al. 2018], and motion segmentation [Brox and Malik 2010].

Transformer architectures have proven extremely adept at modeling long-term relationships within an input sequence via attention mechanisms. Originally used for language tasks [Vaswani et al. 2017; Devlin et al. 2018; Brown et al. 2020], they have since been adapted for solving popular computer vision problems such as image classification [Dosovitskiy et al. 2021], generation [Chen et al. 2020; Ramesh et al. 2021], video understanding [X. Wang et al. 2018; Girdhar et al. 2019; Bertasius et al. 2021], object detection [Carion et al. 2020], and zero-shot classification [Radford et al. 2021]. In this work, we take inspiration from a specific variant of self-attention, namely slot attention [Locatello et al. 2020], which was demonstrated to be effective for learning object-centric representations on synthetic datasets, e.g. CLEVR.

Layered representations were originally proposed by Wang and Adelson [J. Y. Wang and Adelson 1994] to represent a video as a composition of layers with simpler motions. Since then, layered representations of images and videos have been widely adopted in computer vision [Brostow and Essa 1999; Jojic et al. 2001; Zitnick et al. 2004; Pawan Kumar et al. 2008; Xue et al. 2015], often to estimate optical flow [Sun et al. 2012; Sun et al. 2013; Wulff and Black 2014; Wulff and

Black 2015]. More recently, deep learning-based layer decomposition methods have been used to infer depth for novel view synthesis [Tinghui Zhou et al. 2018; Srinivasan et al. 2019], separate reflections and other semi-transparent effects [Alayrac et al. 2019b; Alayrac et al. 2019a; Gandelsman et al. 2019; Lu et al. 2020], or perform foreground/background estimation [Gandelsman et al. 2019]. These works operate on RGB inputs and produce RGB layers, whereas we propose a layered decomposition of optical flow inputs for unsupervised moving object discovery.

Object-centric representations interpret scenes with "objects" as the basic building block (instead of individual pixels), which is considered an essential step towards human-level generalization. There is a rich literature on this topic, for example, in [Greff et al. 2019], the IODINE model uses iterative variational inference to infer a set of latent variables recurrently, with each representing one object in an image. Similarly, MONet [Burgess et al. 2019] and GENESIS [Engelcke et al. 2020] also adopt the multiple encode-decode steps. In contrast, [Locatello et al. 2020] proposes Slot Attention, which enables single step encoding-decoding with iterative attention. However, all of the works mentioned above have only shown applications for synthetic datasets, e.g. CLEVR [Johnson et al. 2017]. In this paper, we are the first to demonstrate its use for object segmentation of realistic videos by exploiting motion, where the challenging nuances in visual appearance (e.g. the complex background textures) have been removed.

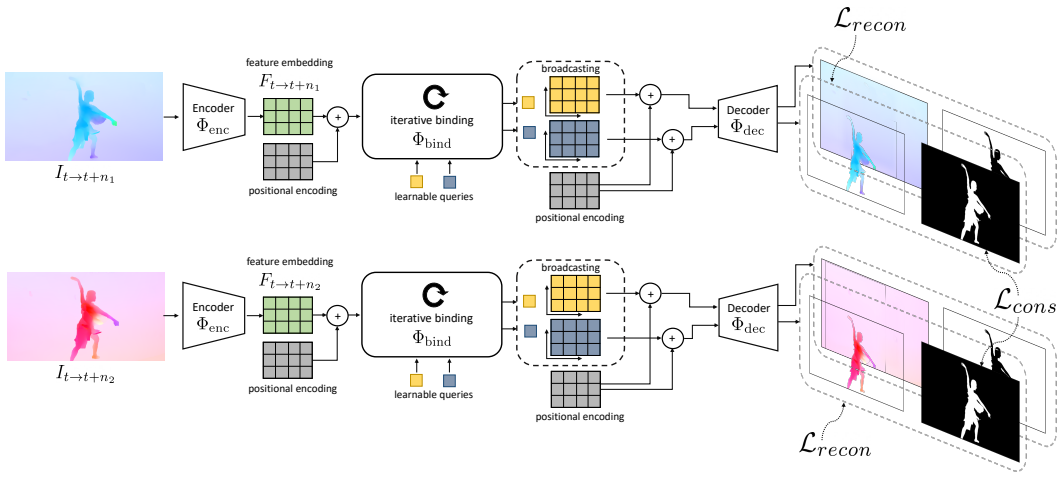


Figure 4.2.1: **Pipeline.** Our model takes optical flow $I_{t\to t+n}$ as input, and outputs a set of reconstruction and opacity layers. Specifically, it consists of three components: feature encoding, iterative binding, and decoding to layers, which are combined to reconstruct the input flow. To resolve motion ambiguities (small motion), or noise in optical flow, consistency between two flow fields computed under different frame gaps is enforced during training. At inference time, only the top half of the figure is used to predict masks from a single-step flow.

4.3 Method

Our goal is to take an input optical flow frame and predict a segment containing the moving object. We aim to train this model in a self-supervised manner, specifically, we adopt an autoencoder-like framework. Our model outputs two layers: one representing the background, and the other for one or more moving objects in the foreground, as well as their opacity layers (weighted masks). Formally, we have:

$$\{\hat{I}^i_{t\to t+n}, \alpha^i_{t\to t+n}\}_{i=1}^N = \Phi(I_{t\to t+n}) \tag{4.1}$$

where $I_{t\to t+n}$ refers to the t to $t+n$ input flow (backward flow when $n<0$), $\Phi(\cdot)$ is the parametrized model, $\hat{I}^i_{t\to t+n}$ is the ith layer reconstruction, $\alpha^i_{t\to t+n}$ is its mask, and $N=2$ is the number of layers (foreground and background). These layers can then be composited linearly to reconstruct the input image $I_{t\to t+n}$:

$$\hat{I}_{t\to t+n} = \sum_{i=1}^{N} \alpha^i_{t\to t+n} \hat{I}^i_{t\to t+n} \tag{4.2}$$

4.3.1 Flow Segmentation Architecture

For simplicity, we first consider the case of a single flow field as input (depicted in the top part of Figure 4.2.1). The entire model consists of three components: (1) a CNN encoder to extract a compact feature representation, (2) an iterative binding module with learnable queries that plays a similar role as soft clustering, i.e. assigning each pixel to one of motion groups, and (3) a CNN decoder that individually decodes each query to full resolution layer outputs (where thresholding the alpha channel yields the predicted segment).

CNN Encoder. We first pass the precomputed optical flow field between two frames, $I_{t\to t+n} \in \mathcal{R}^{3\times H_0\times W_0}$, to a CNN encoder Φ_{enc}, which outputs a lower-resolution feature map:

$$F_{t\to t+n} = \Phi_{\text{enc}}(I_{t\to t+n}) \in \mathcal{R}^{D\times H\times W} \tag{4.3}$$

where H_0, W_0 and H, W refer to the spatial dimensions of the input and output feature maps respectively. Note that, we convert the optical flow field into a three-channel image according to the traditional method specified in the optical flow literature [Sun et al. 2018].

Iterative Binding. The iterative binding module Φ_{bind} aims to group image regions into single entities based on their similarities in motion, i.e. pixels moving in the same direction at the same rate should be grouped together. Intuitively, such a binding process requires a data-dependent parameter updating mechanism, iteratively adapting and enriching the model, gradually including more pixels undergoing similar motions.

To accomplish this task, we adopt a simple variant of slot attention [Locatello et al. 2020], where instead of Gaussian-initialized slots, we use learnable query vectors. Slot attention has recently shown remarkable performance for object-centric representation learning, where the query vectors compete to explain parts of the inputs via a softmax-based attention mechanism, and the representations of these slots are iteratively updated with a recurrent update function. In our case of motion segmentation, ideally, the final representation in each query vector separately

encodes the moving object or the background, which can then be decoded and combined to reconstruct the input flow fields.

Formally, our inputs to Φ_{bind} are feature maps $F_{t\to t+n}$ and two learnable queries (representing foreground and background) $Q \in \mathcal{R}^{D\times 2}$. Learnable spatial positional encodings are summed with $F_{t\to t+n}$; with some abuse of notation, we still refer to this sum as $F_{t\to t+n}$. We use *three* different linear transformations to generate the *query* , *key* and *value*: $q \in \mathcal{R}^{D\times 2}$, $k, v \in \mathcal{R}^{D\times HW}$,

$$q,\ k,\ v = W^Q \cdot Q,\ W^K \cdot F_{t\to t+n},\ W^V \cdot F_{t\to t+n} \tag{4.4}$$

where $W^Q, W^K, W^V \in \mathcal{R}^{D\times D}$.

In contrast to the standard Transformer [Vaswani et al. 2017], the coefficients in slot attention are normalized over all slots. This choice of normalization introduces competition between the slots to explain parts of the input, and ensures each pixel is assigned to a query vector:

$$\begin{aligned} \text{attn}_{i,j} &:= \frac{e^{M_{i,j}}}{\sum_l e^{M_{i,l}}} \\ M &:= \frac{1}{\sqrt{D}} k^T \cdot q,\ \ \text{attn} \in \mathcal{R}^{HW\times 2} \end{aligned} \tag{4.5}$$

To aggregate the input values to their assigned query slot, a weighted mean is used as follows:

$$\begin{aligned} U &:= v \cdot A \ \ \in \mathcal{R}^{D\times 2} \\ \text{where,}\ \ A_{i,j} &:= \frac{\text{attn}_{i,j}}{\sum_l \text{attn}_{l,j}} \end{aligned} \tag{4.6}$$

To maintain a smooth update of the query slots Q, the aggregated vectors U are fed into a recurrent function, parametrized with Gated Recurrent Units (GRU),

$$Q := \text{GRU}(\text{inputs} = U, \text{states} = Q) \tag{4.7}$$

This whole binding process is then iterated T times. The pseudocode can be found in Appendix 4.A.2.

CNN Decoder. The CNN decoder Φ_{dec} individually decodes each of the slots

to outputs of original resolution ($\{\hat{I}^i_{t\to t+n}, \alpha^i_{t\to t+n}\} \in \mathcal{R}^{4\times H_0\times W_0}$), which includes an (unnormalized) single-channel alpha mask and the reconstructed flow fields. Specifically, the input to the decoder is the slot vector broadcasted onto a 2D grid augmented with a learnable spatial positional encoding.

Reconstruction. Once each slot has been decoded, we apply softmax to the alpha masks across the slot dimension, and use them as mixture weights to obtain the reconstruction $\hat{I}_{t\to t+n}$ (Eq. 4.2). Our reconstruction loss is an L2 loss between the input and reconstructed flow,

$$\mathcal{L}_{recon} = \frac{1}{\Omega}\sum_{p\in\Omega} |I_{t\to t+n}(p) - \hat{I}_{t\to t+n}(p)|^2 \tag{4.8}$$

where p is the pixel index, and Ω is the entire spatial grid.

Entropy Regularization. We impose a pixel-wise entropy regularisation on inferred masks:

$$\begin{aligned}\mathcal{L}_{entr} = \frac{1}{\Omega}\sum_{p\in\Omega}(&-\alpha^1_{t\to t+n}(p)\log\alpha^1_{t\to t+n}(p) \\ &-\alpha^0_{t\to t+n}(p)\log\alpha^0_{t\to t+n}(p))\end{aligned} \tag{4.9}$$

This loss is zero when the alpha channels are one-hot, and maximum when they are of equal probability. Intuitively, this helps encourage the masks to be binary, which aligns with our goal in obtaining segmentation masks.

Instance Normalisation. In the case of motion segmentation, objects can only be detected if they undergo an independent motion from the camera; thus previous work attempts to compensate for camera motion [Bideau and Learned-Miller 2016b; Lamdouar et al. 2020]. We are inspired by these ideas, but instead of explicitly estimating homography or camera motion, we take a poor-man's approach by simply using Instance Normalisation (IN) [Ulyanov et al. 2016] in the CNN encoder and decoder, which normalizes each channel in each training sample independently. Intuitively, the *mean* activation tends to be dominated by the motions in the large homogeneous region, which is usually the background. This normalization, in combination with ReLU activations, helps gradually separate the

background motion from the foreground motions.

4.3.2 Self-supervised Temporal Consistency Loss

The segmentation computed for the current frame should be identical irrespective of whether the 'second' frame is consecutive, or earlier or later in time. We harness this constraint to form a self-supervised temporal consistency loss by first defining a set of 'second' frames, and then requiring consistency between their pairwise predictions. We describe the set first, followed by the loss.

Multi-step flow. As objects may be static for some frames, we make our predictions more robust by leveraging observations from multiple timesteps. We consider the flow fields computed from various temporal gaps as an input set, i.e. $\{I_{t\to t+n_1}, I_{t\to t+n_2}\}$, $n_1, n_2 \in \{-2, -1, 1, 2\}$, and use a permutation invariant consistency loss to encourage the model to predict the same foreground/background segmentation for all flow fields in the set.

Consistency loss. We randomly sample two flow fields from the input set and pass them through the model ($\Phi(\cdot)$), outputting the flow reconstruction and alpha masks for each. As the reconstruction loss is commutative, it is not guaranteed that the same slot will always output the background layer; therefore, we use a permutation-invariant consistency loss, i.e. only backpropagating through the lowest-error permutation:

$$\mathcal{L}_{cons} = \frac{1}{\Omega}\min\Big(\sum_{p\in\Omega} |\alpha^1_{t\to t+n_1}(p) - \alpha^1_{t\to t+n_2}(p)|^2, \sum_{p\in\Omega} |\alpha^1_{t\to t+n_1}(p) - \alpha^0_{t\to t+n_2}(p)|^2\Big)$$

Note that, this consistency enforcement only occurs during training. At inference time, a single-step flow is used, as shown in the top half of Figure 4.2.1.

Total Loss. The total loss for training the architecture is:

$$\mathcal{L}_{total} = \gamma_r \mathcal{L}_{recon} + \gamma_c \mathcal{L}_{cons} + \gamma_e \mathcal{L}_{entr} \tag{4.10}$$

we use $\gamma_r = 10^2$, $\gamma_c = 10^{-2}$ and $\gamma_e = 10^{-2}$. However, we found the model is fairly robust to these hyper-parameters.

4.3.3 Discussion

Differences from slot attention. Slot attention was originally introduced for self-supervised object segmentation for RGB images [Locatello et al. 2020], and its usefulness was demonstrated on synthetic data (CLEVR [Johnson et al. 2017]), where objects are primitive shapes with simple textures. However, this assumption is unlikely to hold in the case of natural images or videos, making it challenging to generalize such object-centric representations.

In this work, we build on the insight that although objects in images may not be naturally textureless, their motions typically are. Hence, we develop the self-supervised object segmentation model by exploiting their optical flows, where the nuance in visual appearance is discarded, thus not restricted to simple synthetic cases. As an initial trial, we experimented with the same setting as [Locatello et al. 2020], where query vectors are sampled from a Gaussian distribution; however, we were unable to train it. Instead, we use a learnable embedding here, which we highlight as one of the architectural changes critical to our model's success. Other critical changes include instance normalization and temporal consistency, which we demonstrate in ablations in Section 4.5.1.

Why does it work for motion segmentation? The proposed idea can be seen as training a generative model to segment the flow fields. With the layered formulation, reconstruction is limited to be a simple *linear* composition of layer-wise flow, decoded from a single slot vector.

Conceptually, this design has effectively introduced a representational bottleneck, encouraging each slot vector to represent minimal information, i.e. homogeneous motion, and with minimal redundancy (mutual information) between slots. All these properties make such an architecture well-suited to the task of segmenting objects undergoing independent motions.

4.4 Experimental Setup

In this section, we describe the datasets and evaluation metrics, followed by implementation details.

4.4.1 Datasets

We benchmark on four different datasets that are commonly used for unsupervised video object segmentation.

DAVIS2016 [Perazzi et al. 2016] contains a total of 50 sequences (30 for training and 20 for validation), depicting diverse moving objects such as animals, people, and cars. The dataset contains 3455 1080p frames with pixel-wise annotations at 480p for the predominantly moving object.

SegTrackv2 [F. Li et al. 2013] contains 14 sequences and 976 annotated frames. Each sequence contains 1-6 moving objects, and presents challenges including motion blur, appearance change, complex deformation, occlusion, slow motion, and interacting objects.

FBMS59 [Peter et al. 2014] consists of 59 sequences and 720 annotated frames (every 20th frame is annotated), which vary greatly in image resolution. Sequences involve multiple moving objects, some of which may be static for periods of time.

Moving Camouflaged Animals (MoCA) [Lamdouar et al. 2020] contains 141 HD video sequences, depicting 67 kinds of camouflaged animals moving in natural scenes. Both temporal and spatial annotations are provided in the form of tight bounding boxes for every 5th frame. Using the provided motion labels (locomotion, deformation, static), we filter out videos with predominantly no locomotion, resulting in 88 video sequences and 4803 frames (this subset will be released).

4.4.2 Evaluation Metrics

We use two different evaluation metrics, depending on whether pixel-wise annotations or bounding boxes are provided.

Segmentation (Jaccard). For DAVIS2016, SegTrackv2 and FBMS59, pixel-wise segmentation is provided; thus we report the standard metric, region similarity ($\mathcal{J}$), computing the mean over the test set. For FBMS59 and SegTrackv2, we follow the common practice [Y. Yang et al. 2019; Jain et al. 2017] and combine multiple objects as one single foreground.

Localization (Jaccard & Success Rate). As the MoCA dataset provides only bounding box annotations, we evaluate for the detection task and report results in the form of detection success rate [Everingham et al. 2010; T. Lin et al. 2014], for varying IoU thresholds ($\tau \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$).

4.4.3 Implementation Details

We evaluate three different approaches for computing optical flow, namely, PWC-Net [Sun et al. 2018], RAFT [Teed and Deng 2020] and ARFlow [L. Liu et al. 2020]; the first two are supervised, while the latter is self-supervised. We extract the optical flow at the original resolution of the image pairs, with the frame gaps $n \in \{-2, -1, 1, 2\}$ for all datasets, except for FBMS59, where we use $n \in \{-6, -3, 3, 6\}$ to compensate for small motion. To generate inputs to the network for training, the flows are resized to 128×224 (and scaled accordingly), converted to 3-channel images with the standard visualization used for optical flow, and normalized to $[-1, 1]$.

In the iterative binding module ($\Phi_{\text{bind}}(\cdot)$), we use two learnable query vectors (as we consider the case of segmenting a single moving object from the background), and choose $T = 5$ iterations (as explained in Section 4.3.1). We adopt a simple VGG-style network for the CNN encoder and decoder, with 3 blocks each; in each block, we use 2 sets of convolutions, instance normalization and ReLU activation. We train with a batch size of 64 images and use the Adam optimizer [Kingma and Ba 2014] with an initial learning rate of 5×10^{-4}, decreasing every 80k iterations. The exact architecture description and training schedule can be found in Appendix 4.A.

4.5 Results

Model	Flow	IN	T	$\mathcal{L}_e$	$\mathcal{L}_c$	DAVIS ($\mathcal{J}\uparrow$)
CIS [Y. Yang et al. 2019]	PWC-Net	–	–	–	–	59.2
Ours-A	PWC-Net	✓	5	✓	✓	63.7
Ours-B	**RAFT**	✓	**5**	✓	✓	**68.3**
Ours-C	ARFlow	✓	5	✓	✓	53.2
Ours-D	RAFT	✓	3	✓	✓	65.8
Ours-E	RAFT	✗	3	✓	✓	63.3
Ours-F	RAFT	✗	5	✓	✓	64.5
Ours-G	RAFT	✓	5	✗	✗	48.0
Ours-H	RAFT	✓	5	✗	✓	60.3
Ours-I	RAFT	✓	5	✓	✗	51.2

Figure 4.5.1: **Ablation studies** on flow extraction methods, instance normalization (IN), grouping iterations (T), entropy regularization ($\mathcal{L}_e$) and set consistency ($\mathcal{L}_c$).

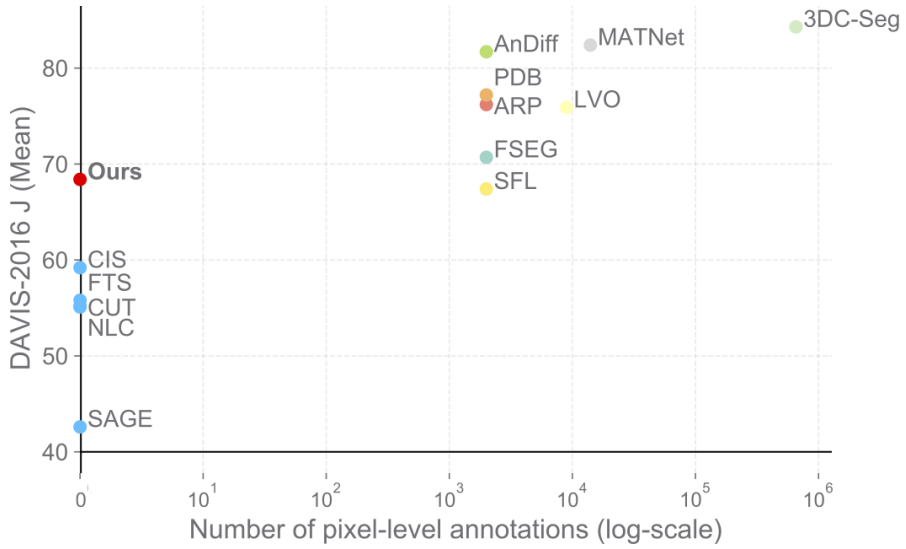


Figure 4.5.2: **Comparison on DAVIS2016.** Note that, supervised approaches may use models pretrained on ImageNet [Deng et al. 2009], but here we only count number of images with pixel-wise annotations.

In this section, we compare primarily with a top-performing approach trained without manual annotations – Contextual Information Separation (CIS [Y. Yang et al. 2019]). However, as the architecture, input resolution, modality and post-processing are all different, we try our best to conduct the comparison as fairly as possible. Note that benchmarks are evaluated at full resolution by simply upsampling the predicted masks.

In order to guarantee a fair comparison of the model, we disregard the gains from post-processing, which typically include averaging an ensemble of multiple crops

Model	RGB	Flow	DAVIS16 ($\mathcal{J}\uparrow$)	STv2 ($\mathcal{J}\uparrow$)	FBMS59 ($\mathcal{J}\uparrow$)	Runtime (sec $\downarrow$)
SAGE	✓	✓	42.6	57.6	**61.2**	0.9s
NLC	✓	✓	55.1	**67.2**	51.5	11s
CUT	✓	✓	55.2	54.3	57.2	103s
FTS	✓	✓	55.8	47.8	47.7	0.5s
CIS	✓	✓	59.2 (71.5)	45.6 (62.0)	36.8 (63.5)	0.1s (11s)
Ours	✗	✓	**68.3**	58.6	53.1	**0.012s**
SFL	✓	✓	67.4	–	–	7.9s
FSEG	✓	✓	70.7	61.4	68.4	–
LVO	✓	✓	75.9	57.3	65.1	–
ARP	✓	✓	76.2	57.2	59.8	74.5s
COSNet	✓	✗	80.5	–	75.6	–
MATNet	✓	✓	82.4	–	–	0.55s
3DC-Seg	✓	✓	84.3	–	–	0.84s

Table 4.5.1: **Full comparison on moving object segmentation** (unsupervised video segmentation). We compare our method against: SAGE [W. Wang et al. 2018], NLC [Faktor and Irani 2014], CUT [Keuper et al. 2015], FTS [Papazoglou and Ferrari 2013], CIS [Y. Yang et al. 2019], SFL [J. Cheng et al. 2017], FSEG [Jain et al. 2017], LVO [Tokmakov et al. 2019], ARP [Song et al. 2018], COSNet [Xiankai et al. 2019], MATNet [Tianfei Zhou et al. 2020], 3DC-Seg [S. Mahadevan et al. 2020]. We consider three popular datasets, DAVIS2016, SegTrack-v2 (STv2), and FBMS59. Models above the horizontal dividing line are trained without using *any* manual annotation, while models below are pre-trained on image or video segmentation datasets, e.g. DAVIS, YouTube-VOS, thus requiring ground truth annotations at training time. Numbers in parentheses denote the additional usage of significant post-processing, e.g. multi-step flow, multi-crop, temporal smoothing, CRFs.

and flow steps, temporal smoothing, and CRFs, at the expense of runtime. Here, we consider only raw framewise predictions with single-step flow at over 80fps, which is more likely to be the application of motion segmentation in practice.

4.5.1 Ablation Studies

We conduct all ablation studies on DAVIS2016, and vary one variable each time, as shown in Table 4.5.1.

Choice of Optical Flow Algorithm. With the same flow extraction method (PWC-Net), our proposed model (Ours-A) outperforms CIS by about 4.5 points on mean Jaccard ($\mathcal{J}$), and using improved optical flow (RAFT) provides further performance gains. We therefore use RAFT from hereon.

Instance Normalization and Grouping. We observe two phenomena: *first*, when holding constant the number of grouping iterations T (3 or 5), models trained

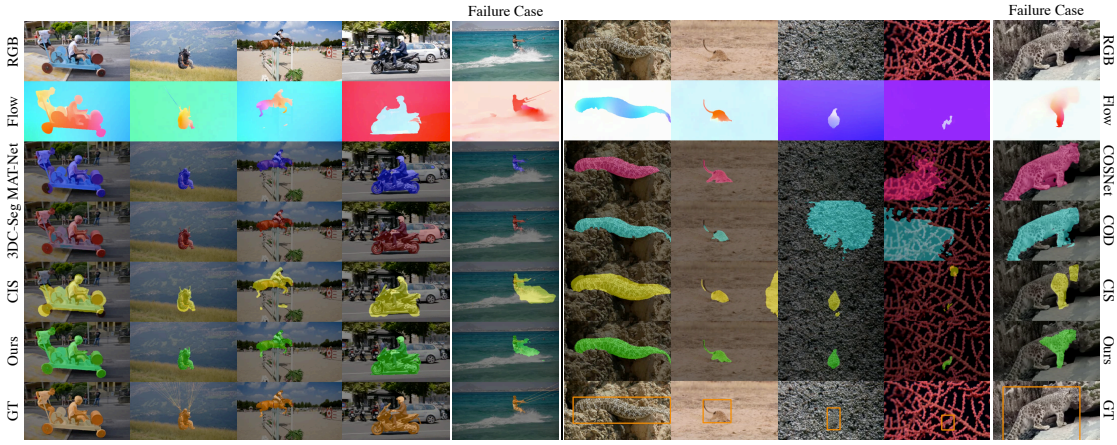


Figure 4.5.3: **Qualitative results.** On DAVIS2016 (left), our method is able to segment a variety of challenging objects such as people, animals, and vehicles, often on-par with top supervised approaches. On MoCA (right), our model is able to accurately segment well-camouflaged objects even when previous supervised methods fail completely (3rd, 4th columns). We show a failure case (left) where the splash created by the person is incorrectly included in our predicted segment, and another failure case (right) where the animal is only partially moving and thus partially segmented.

with instance normalization perform consistently better; *second*, iterative grouping with $T = 5$ is better than that trained with $T = 3$. However, at $T = 8$, the model did not converge in the same number of training steps, and thus we do not include it in the table. For the remainder of the experiments, we use instance normalization and $T = 5$ to balance performance and training time.

Consistency and Entropy Regularization. While comparing Ours-B and Ours-I, we observe that the performance degrades significantly without the temporal consistency loss, and that the entropy regularization is also important, as shown by Ours-B and Ours-H.

4.5.2 Comparison with State-of-the-art

We show comparison results for different datasets in Table 4.5.1. On DAVIS2016, we improve upon the state-of-the-art for unsupervised methods (CIS) by a large margin (+9.1%). As shown in Figure 4.5.2, although we do not use any pixel-level annotations during training, our method is nearing the performance of supervised models trained on thousands of images. In addition, we argue that, motion segmentation in realistic scenarios, e.g. by predator or prey, is likely to require fast processing. Our model operates on small resolution (potentially sacrificing some

				Success Rate					
Model	RGB	Flow	$\mathcal{J}\uparrow$	$\tau = 0.5$	$\tau = 0.6$	$\tau = 0.7$	$\tau = 0.8$	$\tau = 0.9$	SR_{mean}
COD	✗	✓	44.9	0.414	0.330	0.235	0.140	0.059	0.236
COD (two-stream)	✓	✓	55.3	0.602	0.523	0.413	0.267	0.088	0.379
COSNet	✓	✗	50.7	0.588	0.534	0.457	0.337	0.167	0.417
MATNet	✓	✓	**64.2**	**0.712**	**0.670**	**0.599**	**0.492**	**0.246**	**0.544**
CIS	✓	✓	49.4	0.556	0.463	0.329	0.176	0.030	0.311
CIS (post-processing)	✓	✓	54.1	0.631	0.542	0.399	0.210	0.033	0.363
Ours	✗	✓	**63.4**	**0.742**	**0.654**	**0.524**	**0.351**	**0.147**	**0.484**

Table 4.5.2: **Comparison results on MoCA dataset.** We report the successful localization rate for various thresholds τ (see Section 4.4.2). We compare our method against COD [Lamdouar et al. 2020], COSNet [Xiankai et al. 2019], MATNet [Tianfei Zhou et al. 2020], CIS [Y. Yang et al. 2019]. Both CIS and Ours were pre-trained on DAVIS and finetuned on MoCA in a self-supervised manner. Note that, our method achieves comparable Jaccard ($\mathcal{J}$) to MATNet (2nd best model on DAVIS), without using RGB inputs and without any manual annotation for training.

accuracy) with over 80fps.

For SegTrackv2 and FBMS59, they occasionally include multiple objects in a single video, and only a subset of them are moving, making it challenging to spot all objects using flow-only input, but we achieve competitive performance nonetheless. We discuss this limitation below.

4.5.3 Camouflage Breaking

In addition to evaluating on moving object segmentation, we also benchmark the model on camouflaged object detection on the MoCA dataset, where visual cues are often less effective than motion cues. To compare fairly with CIS [Y. Yang et al. 2019], we use the code and model released by the authors, and finetune their model on MoCA in a self-supervised manner. We convert our model's output segmentation mask to a bounding box by keeping only the largest connected region in the prediction before taking the bounding box around it.

We report quantitative results in Table 5.6.4 and show qualitative results in Figure 4.5.3. Our model significantly outperforms CIS (14% when allowing no post-processing), previous supervised approaches e.g. COD [Lamdouar et al. 2020] (18.5% on Jaccard), and even COSNet [Xiankai et al. 2019] (among the top supervised approaches on DAVIS). We conjecture that COSNet's weaker performance is due

to its sole reliance on visual appearance (which is distracting for the MoCA data) rather than using motion inputs. This is particularly interesting, as it clearly indicates that no single information cue is able to do the task perfectly, echoing the two-stream hypothesis [Goodale and Milner 1992] that both appearance and motion are essential to visual systems.

4.5.4 Limitations

Despite showing remarkable improvements on motion segmentation in accuracy and runtime, we note the following limitations of the proposed approach (shown in Figure 4.5.3) and treat them as future work: *first*, the existing benchmarks are mostly limited to motion segmentation into foreground and background, thus, we choose to use two slots in this paper; however, in real scenarios, videos may contain multiple independently moving objects, which the current model will assign to a single layer. It may be desirable to further separate these objects into different layers. *Second*, we have only explored motion-only (optical flow) as input, which significantly limits the model in segmenting objects when flow is uninformative or incomplete (as in Fig. 4.5.3, right); however, the self-supervised video object segmentation objective is applicable also to a two-stream approach, and so RGB could be incorporated. *Third*, the current method may fail when optical flow is noisy or low-quality (Fig. 4.5.3, left); jointly optimizing flow and segmentation is a possible way forward in this case.

4.6 Conclusions

In this paper, we present a self-supervised model for motion segmentation. The algorithm takes only flow as input, and is trained without any manual annotation, surpassing previous self-supervised methods on public benchmarks such as DAVIS2016, narrowing the gap with supervised methods. On the more challenging camouflage dataset (MoCA), our model actually compares favourably to the top approaches in video object segmentation that are trained with heavy supervision. As computation power grows and more high quality videos become available, we believe that self-supervised learning algorithms can serve as a strong competitor to the supervised counterparts for their scalability and generalizability.

Appendix

4.A Training Details

In this section, we outline the training settings and architecture details used in our paper.

4.A.1 Encoder & Decoder

The backbone of network architecture for the model are shown in Table 4.A.1, we refer the readers to the pseudocode for iterative binding module.

	stage	operation	output sizes
	input	–	$3 \times 128 \times 224$
Encoder	conv1	$[5 \times 5, 64] \times 2$	$64 \times 128 \times 224$
	mp1	maxpool, stride $= 2$	$64 \times 64 \times 112$
	conv2	$[5 \times 5, 128] \times 2$	$128 \times 64 \times 112$
	mp2	maxpool, stride $= 2$	$128 \times 32 \times 56$
	conv3	$[5 \times 5, 256] \times 2$	$256 \times 32 \times 56$
Decoder	conv^T1	$5 \times 5, 64, \text{stride} = 2$	$64 \times 2 \times 32 \times 56$
	conv^T2	$5 \times 5, 64, \text{stride} = 2$	$64 \times 2 \times 64 \times 112$
	conv^T3	$5 \times 5, 64, \text{stride} = 2$	$64 \times 2 \times 128 \times 224$
	outconv	$5 \times 5, 64,$ $5 \times 5, 4$	$4 \times 2 \times 128 \times 224$

Table 4.A.1: Network architecture. All convolutions have padding 2 to preserve spatial resolution, and are followed by instance normalization and ReLU activation, except the final layer. The details of the iterative binding module is in Figure 4.A.1.

4.A.2 Iterative Binding Module

The pseudocode for the iterative binding module is shown in Figure 4.A.1.

Algorithm: Pseudo Code for Iterative Binding / Grouping

```
# inputs: feature maps + position encoding
slots = Embedding(D, 2)    # [bsz, 2, D]
K = project_K(inputs)      # [bsz, HW, D]
V = project_V(inputs)      # [bsz, HW, D]
for _ in range(T):         # T iterations
    slots = LayerNorm(slots) # [bsz, 2, D]
    Q = project_Q(slots)     # [bsz, 2, D]
    scores = dot(K, Q.t())   # [bsz, HW, 2]
    attn = softmax(scores / sqrt(D), axis=-1) # [bsz, HW, 2]
    updates = weighted_mean(attn.t(), V)      # [bsz, 2, D]
    slots = GRU(slots, updates)               # [bsz, 2, D]
    slots = slots + MLP(LayerNorm(slots))     # [bsz, 2, D]
```

Figure 4.A.1: Pseudocode for iterative binding. The linear projections for key (K) and value (V) have 256 dimensions. The MLP has two layers, both with 256 dimensions, with ReLU in between.

4.A.3 Hyperparameters

For all datasets, we train with batch size of 64, although note that this corresponds to 32 *pairs* of optical flow in order to train with consistency loss. Our initial learning rate is 5e-4, with first 200 steps being warmup, and decays by half every 8e4 iterations. During this decay, the scale for entropy and consistency loss is also increased by a factor of 5, gradually encouraging the predicted alpha channel to be binary. We train the algorithm for about 300k iterations.

4.B MoCA Dataset Curation

The MoCA dataset contains 141 high-definition video sequences, with an average duration of 11 seconds. These sequences were collected from YouTube with resolution 720×1280, and sampled at 24 fps, resulting in 37K frames depicting 67 kinds of camouflaged animals moving in natural scenes. Both temporal and spatial annotations are provided in the form of tight bounding boxes on every 5th frame. We use a modified version of this dataset in order to make it more suitable for

segmentation tasks. We outline the process and rationale below.

- We crop away the channel logos and empty border spaces, and resize the low-resolution images to the same resolution as the other images in the dataset (at 720×1280). We then adjust the ground-truth annotations accordingly.
- The original authors resampled all videos to 24 fps even when some original videos have less, causing some consecutive frames to be identical. To alleviate this, we sample every 3 frames from the original dataset, up to 100 frames per video.
- For annotations, we use linear interpolation to generate the missing frames' bounding boxes, resulting in a dense frame-wise annotation.
- The authors also provided the motion labels for each annotated frame (locomotion, deformation, static), so we filter away videos with predominantly no locomotion.
- We also further discard videos that contain large amount of frames where the motion does not belong to the primary object.

This eventually results in 88 video sequences and 4803 frames, which we will release for fair comparison.

4.C Further Results

We show more qualitative results in Figures 4.C.1 and 4.C.2.



Figure 4.C.1: Qualitative results on DAVIS2016 and MoCA, respectively.



Figure 4.C.2: Qualitative results on SegTrackv2 and FBMS59, respectively.

Chapter 5

The Making and Breaking of Camouflage

This work is under review.

The Making and Breaking of Camouflage

Hala Lamdouar Weidi Xie Andrew Zisserman

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.

5.1 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.*, 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 ex-

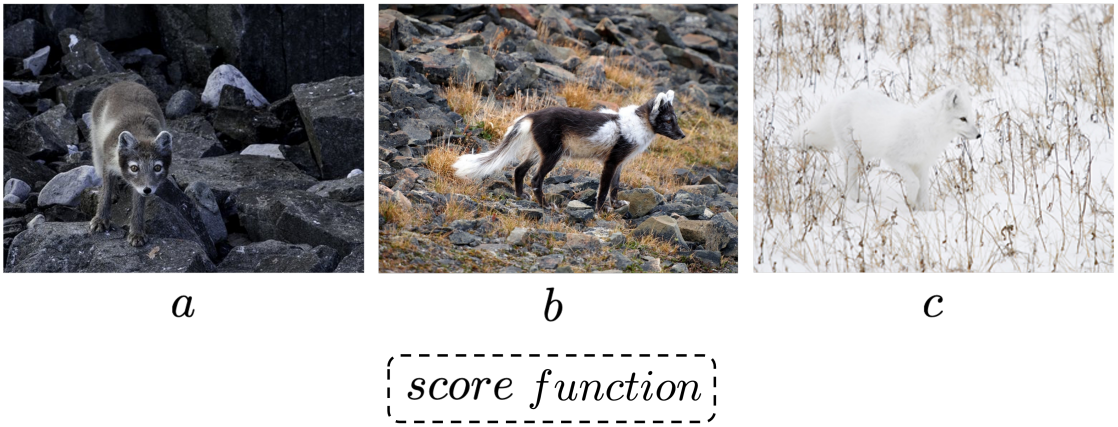


Figure 5.1.1: The three images depict the same animal, an Arctic fox, as it adapts its appearance to better blend with the changing landscape of the new season. While images a and c exhibit better background similarity than image b, the fox boundary is more visible in image a than image c. We assess the effectiveness of camouflage by measuring the degree of ambiguity it creates with respect to its background.

ample, that 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 [Owens et al. 2014; Guo et al. 2022], that tackled camouflage as a problem of optimal texture synthesis that makes 3D objects non-detectable in a given scene. Others have addressed camouflage as a highly challenging object segmentation task in [D. Fan et al. 2021; Le et al. 2016; Mei et al. 2021]. Efforts have been made in collecting large scale camouflage datasets [D. Fan et al. 2021; Le et al. 2016] 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 [D. Fan et al. 2021]. While another line of research turned to camouflage breaking in sequences by taking advantage of motion cues [Lamdouar et al. 2020; Bideau and Learned-Miller 2016b; Lamdouar et al. 2021; C. Yang et al. 2021; X. Cheng et al. 2022], the camouflage data scarcity is even more extreme for videos. Recently, Sim2Real [Lamdouar et al. 2021; J. Xie et al. 2022] training has shown to be very effective for motion segmentation. By training on the optical flows of synthetic videos, the model can generalise to real videos without suffering from

the domain gaps.

In this paper, we take inspiration from these works and propose a synthetic data generation framework for generating examples of highly effective camouflage. To start with, we investigate three measures 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 play 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, for example, they can characterise the inverse measure for the saliency of an object. We propose two camouflage scores, and compute them on existing camouflage datasets, showing that they are highly correlated with human experiments.

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 superior performance boost.

In summary, in this paper, 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 video camouflaged animal segmentation benchmark.

5.2 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 [T. Lin et al. 2014; Troscianko et al. 2017] or conducting subjective perceptual experiments [Owens et al. 2014; Guo et al. 2022; Skurowski et al. 2018]. In [Owens et al. 2014; Guo et al. 2022], 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.* [Skurowski et al. 2018] asked human volunteers to rate CHAMELEON [Skurowski et al. 2018] 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 [Brox and Malik 2010; Ochs and Brox 2011; Keuper et al. 2015; Sivic et al. 2004; Lezama et al. 2011; Ochs and Brox 2012; Ochs et al. 2013]. Another line of research tackled the problem by compensating for the background motion [Bideau and Learned-Miller 2016b; Bideau et al. 2018; Lamdouar et al. 2020], via registering consecutive frames, or explicitly estimating camera motion. More recently, in [C. Yang et al. 2021], a Transformer-like architecture is used to reconstruct the input flows, and the segmentation masks is generated as a side product, while [Choudhury et al. 2022] decomposes the flow filed into regions by fitting affine transformations. The mostly closely related to ours is [Lamdouar et al. 2021; J. Xie et al. 2022], 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* video sequences.

Fréchet Inception Distance and its variants. Also known as Wasserstein-2 distance [Villani 2009], Fréchet distance (FD) is a metric quantifying the difference between two probability distributions [Fréchet 1957]. Lately, [Heusel et al. 2017] 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 Incep-

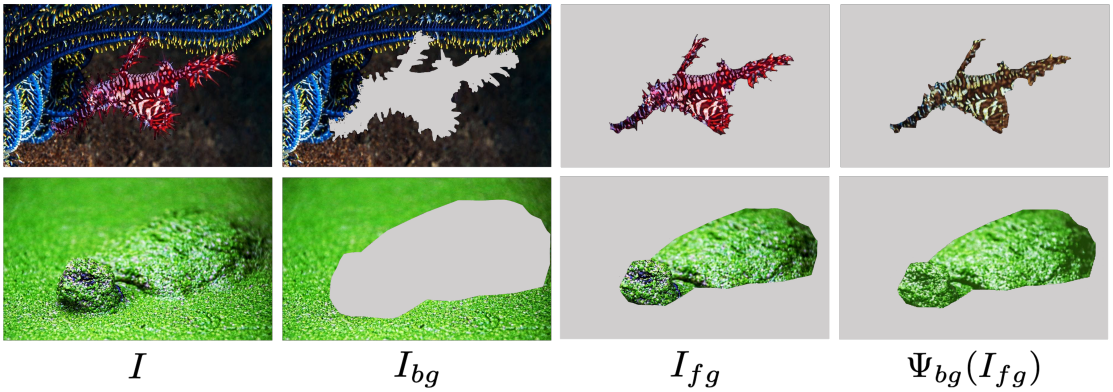


Figure 5.2.1: The reconstruction fidelity score S_{R_f} evaluates the similarities between the original foreground I_{fg} and its reconstruction from background features $\Psi_{bg}(I_{fg})$. The top example shows a case where the animal exhibits different visual patterns (color) from its background S_{R_f}=0.11, while the bottom example shows a better background matching S_{R_f}=0.82.

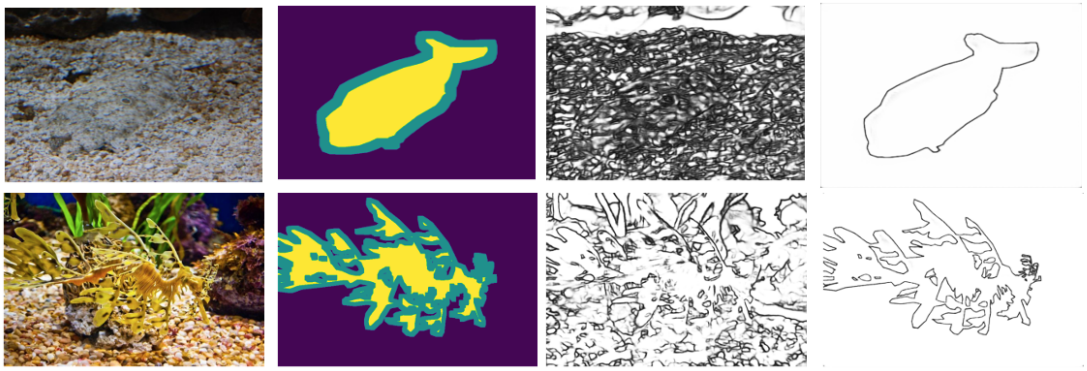


Figure 5.2.2: Trimap regions and animal contours. From left to right: Input image; Trimap partition (foreground region in yellow, background in purple, and boundary region in green); Contours C computed on the original image; and groundtruth Contours C_{GT}. The top example shows less contour agreement along the boundary region (S_b=0.72) than the bottom example (S_b=0.40).

tionV3 [Szegedy et al. 2016]. Following their steps, [Shaham et al. 2019] adapted FID to the image re-targeting task and introduced the single image SIFID Fréchet Inception Distance. Instead of comparing entire image datasets, they compare two images and consider the distributions of their feature vectors obtained from earlier layer of InceptionV3 [Szegedy et al. 2016]. Recently, [Doan et al. 2020] included FID in the training objective and introduced Fréchet-GAN. Our work takes inspiration from [Shaham et al. 2019; Doan et al. 2020] and investigates a FID of regions within the same image as an auxiliary loss.

5.3 Measuring the Effectiveness of Camouflage

Assuming there exists a dataset, $\mathcal{D} = \{(I_1, m_1), \ldots, (I_N, m_N)\}$, 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.

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

Specifically, for scenario with $i \neq j$ 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 the 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 two such scoring functions, specifically, we exploit the animal's mask and quantify the key aspects that determine the camouflage success both perceptually (Sec. 5.3.1) and probabilistically (Sec. 5.3.2). 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 surround of the animal. Hence, when referring to background, we mean the local background and only consider the cropped images centred around the object.

5.3.1 Perceptual Camouflage Score

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

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, if camouflage is successful, we should be able to reconstruct the foreground animal perfectly by simply 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,

$$I_{\text{fg}},\ I_{\text{bg}} = m_{\text{fg}} \odot I,\ m_{\text{bg}} \odot I \tag{5.1}$$

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

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 patch-wise similarity is computed by comparing corresponding RGB values, *i.e.*, low-level perceptual information is exploited for similarity matching. This reconstruction method is very much inspired by the texture generation and inpainting approaches, such as [Efros and Leung 1999]. In practice we implement it with fast approximate nearest neighbor search for efficiency.

The reconstruction fidelity score can thus be computed by assessing the difference value between the foreground region and its reconstruction, specifically, we count the number of foreground pixels that have been successfully reconstructed from the background:

$$S_{R_f}(I, m) = \frac{1}{N_{\text{fg}}} \sum_{(i,j) \in I_{\text{fg}}} R_f(i, j) \tag{5.2}$$

$$R_f(i, j) = \begin{cases} 1 & \text{if } ||I_{\text{fg}} - \Psi_{I_{\text{bg}}}(I_{\text{fg}})||_2 < \lambda ||I_{\text{fg}}||_2 \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

$\Psi_{I_{bg}}(.)$ denotes the reconstruction operation, $N_{\text{fg}} = |m_{\text{fg}}|$ is the total number of pixels in the foreground region, and λ is a threshold (λ=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 standout from its surrounding. In Fig. 5.2.1, we present examples of camouflage evaluation by reconstruction fidelity.

Boundary visibility score

With the trimap representation, we can easily extract the boundary region, $I_{\mathrm{b}} = m_{\mathrm{b}} \odot I$, where $m_{\mathrm{b}} = (1 - m_{\mathrm{bg}}) - m_{\mathrm{fg}}$. 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 [Poma et al. 2020], and run it on original image and foreground mask, to generate the images' contour (C) and groundtruth animal's contour (C_{gt}), as shown in Fig. 5.2.2.

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

$$S_b(I, m) = 1 - \mathrm{F1}(m_b \odot C_{\mathrm{gt}},\ m_b \odot C) \tag{5.4}$$

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:

$$S_\alpha = (1 - \alpha) S_{R_f} + \alpha S_b \tag{5.5}$$

We will describe our method for setting the weighting parameter α in the Experiment section.

5.3.2 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:

$$f_{fg},\ f_{bg} = \Phi_{\mathrm{v}}(I_{fg}),\ \Phi_{\mathrm{v}}(I_{bg}) \tag{5.6}$$

Here $\Phi_{\mathrm{v}}(\cdot)$ denotes a pre-trained Inception network [Szegedy et al. 2015; Szegedy et al. 2016], and we take the feature maps from the early layers. We take inspiration from [Shaham et al. 2019], 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: $f_{\mathrm{fg}} \sim \mathcal{N}(\mu_{\mathrm{fg}}, \Sigma_{\mathrm{fg}})$ and $f_{\mathrm{bg}} \sim \mathcal{N}(\mu_{\mathrm{bg}}, \Sigma_{\mathrm{bg}})$. The intra-image Fréchet distance can be formulated as follows:

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

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

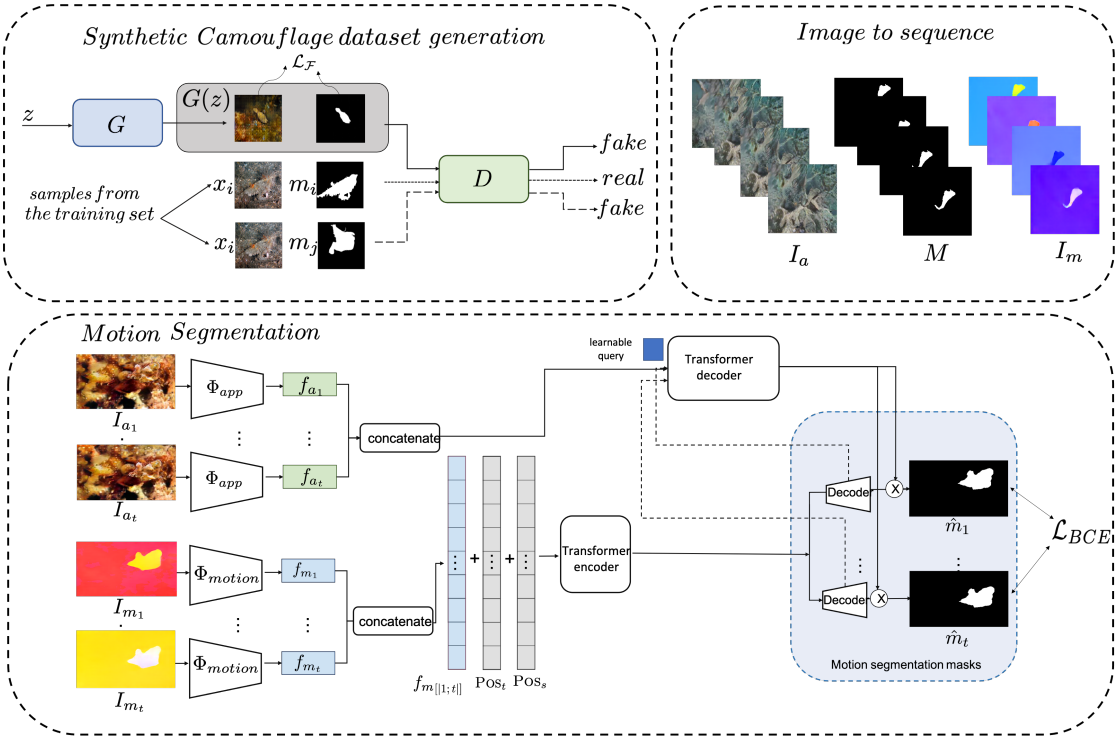


Figure 5.3.1: **Top left**: Our framework consists of a synthetic camouflage data generation pipeline where we train a generator G, in a GAN setting, to create camouflage images and masks, while encouraging camouflage effectiveness via minimising $\mathcal{L}_{\mathcal{F}}$. **Top right**: The generated samples are then transformed into synthetic video sequences following our method presented in Sec. 5.4.2.**Bottom**: the transformer-based motion segmentation architecture for camouflage breaking in videos. It is trained on the synthetic video sequences.

5.4 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.

5.4.1 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 [D. Fan et al. 2021] 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, recall the generative adversarial loss functions [Goodfellow et al. 2014]:

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

Here we adopt the unsaturated version of the generator loss $\mathcal{L}_G$ that maximises the probability of assigning incorrect labels, *i.e.* $y = 1$, to $G(z)$.

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 $i \neq j$:

$$\mathcal{L}_{\text{Coh}} = E_{(x_i,m_j)_{i\neq j}}[-\log(1 - D(x_i, m_j))] \tag{5.7}$$

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

$$\mathcal{L}_{\tilde{D}} = \mathcal{L}_D + \mathcal{L}_{\text{Coh}} \qquad \mathcal{L}_{\tilde{G}} = \mathcal{L}_G + \beta d^2_{\mathcal{F}} \tag{5.8}$$

We present an overview of our data generation framework in Fig. 5.3.1.

5.4.2 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 [Lamdouar et al. 2021]. Specifically, we first inpaint the backgrounds with an off-the-shelf model proposed by Suvorov *et al.* [Suvorov et al. 2021] and overlay the synthetic animal (extracted from the original generated image 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 subsequences where the foreground and background undergo the same motion to simulate momentarily static objects. As we train our generator using the intra-image Fréchet loss $\mathcal{L}_{\mathcal{F}}$, the generated images exhibit strong similarities between the foreground animal and the background. Hence, even if the animal is placed at a different location from the original, we expect it to remain concealed within its surrounding.

5.5 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. We build on two previous architectures, namely, the motion segmentation from [Lamdouar et al. 2021] and the Search Identification Network from [D. Fan et al. 2021]. While the first architecture processes sequences of optical flow, the second takes single RGB images as inputs and treats them separately. Specifically, our proposed model takes both the optical flow and RGB frame sequences as input, where the flow is computed with RAFT from the generated RGB sequence, and processes them by motion encoder, appearance encoder, Transformer-based aggregator, then finally decode into segmentation masks.

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

$$\{f_{m_1}, f_{m_2}, .., f_{m_t}\} = \Phi_{\text{motion}}(\mathbf{I_m})$$

Appearance encoder. Here, we adopt a SINet-v2 [D. Fan et al. 2021] architecture, that encodes the RGB sequence, $I_a = \{I_{a_1}, I_{a_2}, .., I_{a_t}\} \in \mathcal{R}^{t \times c_1 \times h \times w}$ into appearance features:

$$\{f_{a_1}, f_{a_2}, .., f_{a_t}\} = \Phi_{\text{app}}(\mathbf{I_a})$$

Aggregation. A transformer encoder takes as input the motion features together

with spatio-temporal positional encoding. Finally, a transformer-based decoder takes the output of the transformer encoder together with the appearance features and predicts the masks of the moving camouflaged animal in a masked attention style [B. Cheng et al. 2022]. Specifically, using a learnable query, a transformer decoder attends to multiple resolutions of the motion features concatenated with the appearance features and produces an attention map that is multiplied with the highest resolution motion features to produce the segmentation masks: $\{\hat{m}_1, \hat{m}_2, .., \hat{m}_t\}$.

Decoder. A light-weight convNet decoder is used with skip-connections to recover high-resolution motion features, which are fed to the transformer-decoder.

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

5.6 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.

5.6.1 Datasets

Here we describe the publicly available camouflage datasets that we included in our experiments, and the synthetic camouflage datasets that we generated.

CHAMELEON [**Skurowski et al. 2018**] 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 [**Le et al. 2016**]**.** 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 pixel-wise annotations provided.

COD10K [**D. Fan et al. 2021**]**.** 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 [Bideau and Learned-Miller 2016b]. This dataset consists of 9 video sequences of camouflage animals from 6 categories.

MoCA [Lamdouar et al. 2020] 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) [C. Yang et al. 2021], and others have provided dense pixel annotation in MoCA-Mask [X. Cheng et al. 2022]. We use the latter version in our experiments.

Camouflaged cuboids[Owens et al. 2014; Guo et al. 2022]. 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 [Guo et al. 2022] from 36 scenes as well as the cuboids masks.

Synthetic Camouflage Images. Using the method described in Sec. 5.4.1, 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. 5.4.2. We split these into 800 sequences for training and 200 for testing.

5.6.2 Implementation details

For our synthetic camouflage image generation, we adopt a Style-GAN architecture [Zhang et al. 2022] 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 the Newton Schultz iterations when calculating the matrix square root term in Intra-Image Fréchet Inception Distance, with $T = 100$ iteration.

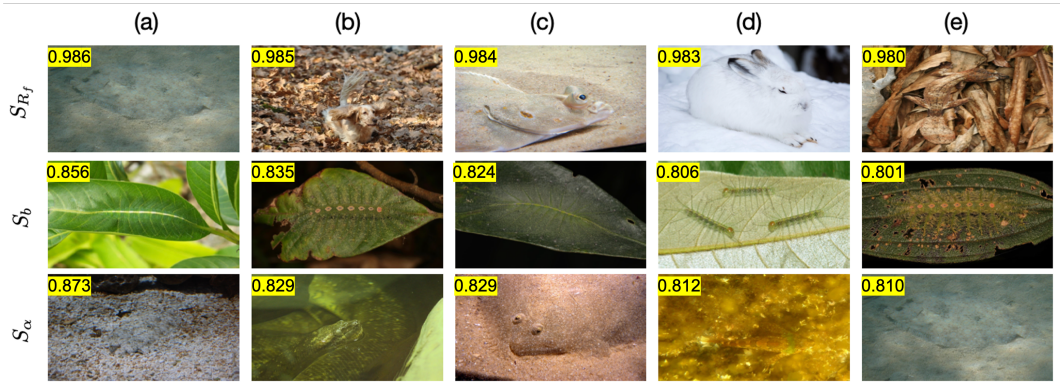


Figure 5.6.1: Top-5 scored examples from COD10K for $S_{Rf}(top)$, $S_b(middle)$ and $S_\alpha(bottom)$. Note, in case you cannot spot the animals, segmentation maps are included in the Appendix 5.7.

When generating the image sequences, we compute optical flows with RAFT [Teed and Deng 2020], and train our moving camouflaged animal segmentation model. We use batches of size 2 and sequences of size $t = 5$. 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.

5.6.3 Results

This section presents qualitative and quantitative results demonstrating the effectiveness of our score functions and the benefit of including $d^2_{\mathcal{F}}$ in the training loop.

Evaluation on camouflage effectiveness

We compute the camouflage score for all the camouflage image and video datasets and report the results in Tab. 5.6.1. For the video datasets, we compute the mean score per sequence, and for the multiple view dataset we report the mean per scene. MoCA-Mask [Lamdouar et al. 2020; X. Cheng et al. 2022] contains the most successful camouflages according to our scoring. While COD10K [D. Fan et al. 2021] subsets are balanced in terms of camouflage effectiveness, we find that, for CAMO [Le et al. 2016], the test subset yields higher scores than the training subset and therefore contains more challenging examples.

Ranking camouflage images in COD10K.

We can rank the images based on the effectiveness of the animal's camouflage, with our proposed scoring functions. As shown in Fig. 5.6.1, we show the five best

Datasets	Data type	$S_{R_f} \uparrow$	$S_b \uparrow$	$S_\alpha \uparrow$	$d^2_{\mathcal{F}} \downarrow$
CHAMELEON	Image	0.694	0.445	0.607	**0.70**
CAMO Train	Image	0.672	0.451	0.595	1.01
CAMO Test	Image	0.683	0.470	0.608	1.00
COD10K Train	Image	0.655	0.433	0.577	0.90
COD10K Test	Image	0.657	0.431	0.578	0.90
Camouflaged Animals	Video	0.674	**0.536**	0.626	1.60
MoCA-Mask Train	Video	**0.850**	0.443	**0.707**	1.14
MoCA-Mask Test	Video	0.733	0.464	0.639	2.51
Camouflaged cuboids	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 5.6.1: Results of the proposed scores on natural camouflage datasets (top): CHAMELEON [Skurowski et al. 2018], CAMO [Le et al. 2016], COD10K [D. Fan et al. 2021], Camouflaged Animals [Bideau and Learned-Miller 2016b], MoCA-Mask [Lamdouar et al. 2020; X. Cheng et al. 2022];and synthetically generated camouflage datasets (bottom): Camouflaged cuboids[Owens et al. 2014; Guo et al. 2022] and our synthetic datasets. 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.

Scores	CHAMELEON	Camouflaged cuboids
S_{R_f}	0.01	-0.07
S_b	0.03	0.42
S_α	**0.42**	**0.41**
$d^2_{\mathcal{F}}$	0.10	-0.17

Table 5.6.2: Kendall-τ metric between rankings produced from the different scoring functions and human scoring groundtruth for CHAMELEON [Skurowski et al. 2018] and Camouflaged cuboids[Owens et al. 2014; Guo et al. 2022] datasets.

Classes	$S_{R_f} \uparrow$	$S_b \uparrow$	$S_\alpha \uparrow$
Aquatic	**0.711**	0.427	**0.612**
Terrestrial	0.599	**0.443**	0.545
Flying	0.668	0.426	0.584
Amphibian	0.678	0.432	0.592
Other	0.606	0.439	0.548
Best	Stingaree(0.885)	Lion (0.502)	Flounder (0.725)
Worst	Ant (0.294)	Stingaree (0.391)	Ant (0.347)

Table 5.6.3: Results per class from COD10K

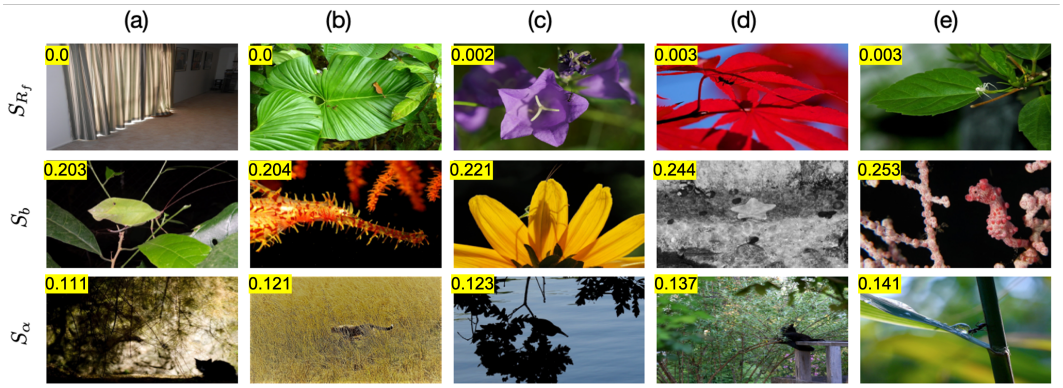


Figure 5.6.2: Lowest scored examples from COD10K for $S_{Rf}(top)$, $S_b(middle)$ and $S_\alpha(bottom)$.

scored images from the large scale COD10K dataset. We can make the following observations: *First*, the top row shows the best examples with respect to S_{R_f} which favours the background matching of the animals without taking into account the boundary region. With the exception of image (a), all examples from the first row exhibit visible boundaries. This is especially noticeable in the rabbit example (d) along the ears and shadow regions. *Second*, the best five 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 stand out as they exhibit colors and patterns that are not present within their background. *Third*, the S_α combining both scores shows highly effective camouflage examples.

In contrast, Fig. 5.6.2 shows the lowest scored images for each approach. For S_{R_f}, we find examples with low background matching in the first row, with three ant examples, *i.e.*, (b), (c), (d). 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.

Comparison with human-produced rankings. We compare our rankings to the human scoring system based on the rating of CHAMELEON [Skurowski et al. 2018] and the time-to-find for Camouflaged cuboids [Guo et al. 2022]. To search for the optimal α parameter in Equation (5.5), we select 15 images from CHAMELEON and compare their S_α ranking with groundtruth using kendall-τ metric [Kendall 1938]. For $\alpha = 0.35$, we obtain kendall-$\tau = 0.51$ on this validation set that we excluded from our test reported in Tab. 5.6.2.

From Tab. 5.6.2, we can draw the following observations: *First*, we find that the boundary score S_b produces more agreement with the groundtruth ranking than S_{R_f}, suggesting that human observers tend to pay more attention to contour visibility than background matching. *Second*, while comparing to the ranking from experiment B [Guo et al. 2022], 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 [Guo et al. 2022], *i.e.*, all the textured cuboids are synthetically generated with very high background matching. *Third*, for both experiments, we obtain the strongest agreement with the S_α score combining both background similarity and boundary blending.

Class analysis. We compute our scores for each of the classes from COD10K and report the results in Tab. 5.6.3. Overall, the highest scored examples, according to our S_α, were from the Aquatic class. We found the Stingaree subclass camouflage to be the most effective in terms of background matching, S_{R_f}=0.885, and the worst in terms of contour visibility. The Best subclass for the boundary score is the Lion S_b=0.502 followed by the caterpillar S_b=0.492.

On the effectiveness of $\mathcal{L}_{\mathcal{F}}$ Adding the intra-image Fréchet Distance 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. 5.6.1. We refer the reader to the Appendix 5.7 for examples of generated samples.

Results on MoCA

We train our model on the synthetically generated camouflage images and fine-tune on the training set of MoCA-Mask. Tab. 5.6.4 presents our camouflage object segmentation results on the test set of MoCA-Mask. Ours-SINet-V2 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. Ours-SINet-V2 outperforms RGB and Motion based methods on MoCA-Mask.

5.6.4 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 camou-

Model	RGB	Motion	mIoU
SINet [D. Fan et al. 2021]	✓		12.8
SINet-V2 [D. Fan et al. 2021]	✓		14.6
SLTNet-Short [X. Cheng et al. 2022]	✓	✓	20.9
SLTNet-long [X. Cheng et al. 2022]	✓	✓	27.6
Motion-group [C. Yang et al. 2021]		✓	12.7
Ours-flow		✓	17.8
Ours-SINet-V2	✓	✓	**30.8**

Table 5.6.4: **Results on MoCA-Mask.**

flage score to curate the generated sequence dataset and filter out the most visible objects.

Our proposed camouflage scores use the groundtruth 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 (b) from the last row of Fig. 5.6.2 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.

5.7 Conclusions

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.

Appendix

We first provide more results of ranking camouflage datasets in Section 5.A and compare the rankings obtained by the different scores in terms of kendall-τ metric [Kendall 1938]. Then in Section 5.B, we present examples of the camouflage image and mask pairs generated via our proposed Generative Adversarial Networks setting with and without training with the $\mathcal{L}_{\mathcal{F}}$ loss term as well as examples of the generated synthetic camouflage sequences. Finally, in Section 5.C, we provide a detailed description of our model's architecture and show qualitative results of our motion segmentation model trained on the synthetic dataset.

5.A Ranking Camouflage Images

We present, in this section, examples of highest and lowest scored images and sequences from the camouflage datasets with respect to the different proposed camouflage assessment scores and we further report their pairwise correlations in terms of the kendall-τ metric [Kendall 1938].

5.A.1 Correlation of the proposed score rankings

In this section, we present the results of comparison of the ranking obtained using our scores in terms of kendall-τ metric, on CAMO [Le et al. 2016] in Tab. 5.A.1, COD10K [D. Fan et al. 2021] in Tab. 5.A.2, MoCA-Mask [Lamdouar et al. 2020; X. Cheng et al. 2022] in Tab. 5.A.3, CHAMELEON [Skurowski et al. 2018] in Tab. 5.A.5, Camouflaged Animals [Bideau and Learned-Miller 2016b] in Tab. 5.A.4. Note that we have inverted the rankings obtained with the $d_{\mathcal{F}}$ distance for consistency with the other score rankings. With our choice for $\alpha = 0.35$, therefore privileging reconstruction fidelity over boundary score, $S_\alpha$is correlated with S_{R_f}. Our proxy score $d^2_{\mathcal{F}}$ shows correlation with $S_\alpha$on COD10k and MoCA-Mask with the highest correlation on Camouflaged Animals $\tau = 0.245$.

Rankings	S_{R_f}	S_b	S_α	$d^2_{\mathcal{F}}$
S_{R_f}	1	0.012	0.035	-0.014
S_b	0.012	1	0.012	0.002
S_α	0.035	0.012	1	-0.040
$d^2_{\mathcal{F}}$	-0.014	0.002	-0.040	1

Table 5.A.1: Kendall-τ of rankings of CAMO [Le et al. 2016]

Rankings	S_{R_f}	S_b	S_α	$d^2_{\mathcal{F}}$
S_{R_f}	1	0.0	0.004	0.003
S_b	0.0	1	0.007	-0.010
S_α	0.004	0.007	1	0.013
$d^2_{\mathcal{F}}$	0.003	-0.010	0.013	1

Table 5.A.2: Kendall-τ of rankings of COD10K [D. Fan et al. 2021]

Rankings	S_{R_f}	S_b	S_α	$d^2_{\mathcal{F}}$
S_{R_f}	1	0.083	0.102	0.097
S_b	0.083	1	0.081	0.057
S_α	0.102	0.081	1	0.091
$d^2_{\mathcal{F}}$	0.097	0.057	0.091	1

Table 5.A.3: Kendall-τ of rankings of MoCA-Mask [Lamdouar et al. 2020; X. Cheng et al. 2022]

Rankings	S_{R_f}	S_b	S_α	$d^2_{\mathcal{F}}$
S_{R_f}	1	0.291	0.349	0.229
S_b	0.291	1	0.287	0.343
S_α	0.349	0.287	1	0.245
$d^2_{\mathcal{F}}$	0.229	0.343	0.245	1

Table 5.A.4: Kendall-τ of rankings of Camouflaged Animals [Bideau and Learned-Miller 2016b]

Rankings	S_{R_f}	S_b	S_α	$d^2_{\mathcal{F}}$
S_{R_f}	1	-0.032	0.067	-0.084
S_b	-0.032	1	-0.057	-0.076
S_α	0.067	-0.057	1	-0.015
$d^2_{\mathcal{F}}$	-0.084	-0.076	-0.015	1

Table 5.A.5: Kendall-τ of rankings of CHAMELEON [Skurowski et al. 2018]

5.A.2 Example of ranked images

We present the rankings of COD10K in Fig. 5.A.1 and Fig. 5.A.2, for MoCA-Mask in Fig. 5.A.3 and Fig. 5.A.4 and for CAMO in Fig. 5.A.5 and Fig. 5.A.6.



Figure 5.A.1: **Top scored examples from COD10K.** From top to bottom: S_{Rf}, S_b, S_α and $d^2_{\mathcal{F}}$ rankings. For each image, we show the corresponding groundtruth mask and the computed score.



Figure 5.A.2: **Lowest scored examples from COD10K.** From top to bottom: S_{Rf}, S_b, S_α and $d^2_{\mathcal{F}}$ rankings. For each image, we show the corresponding groundtruth mask and the computed score.

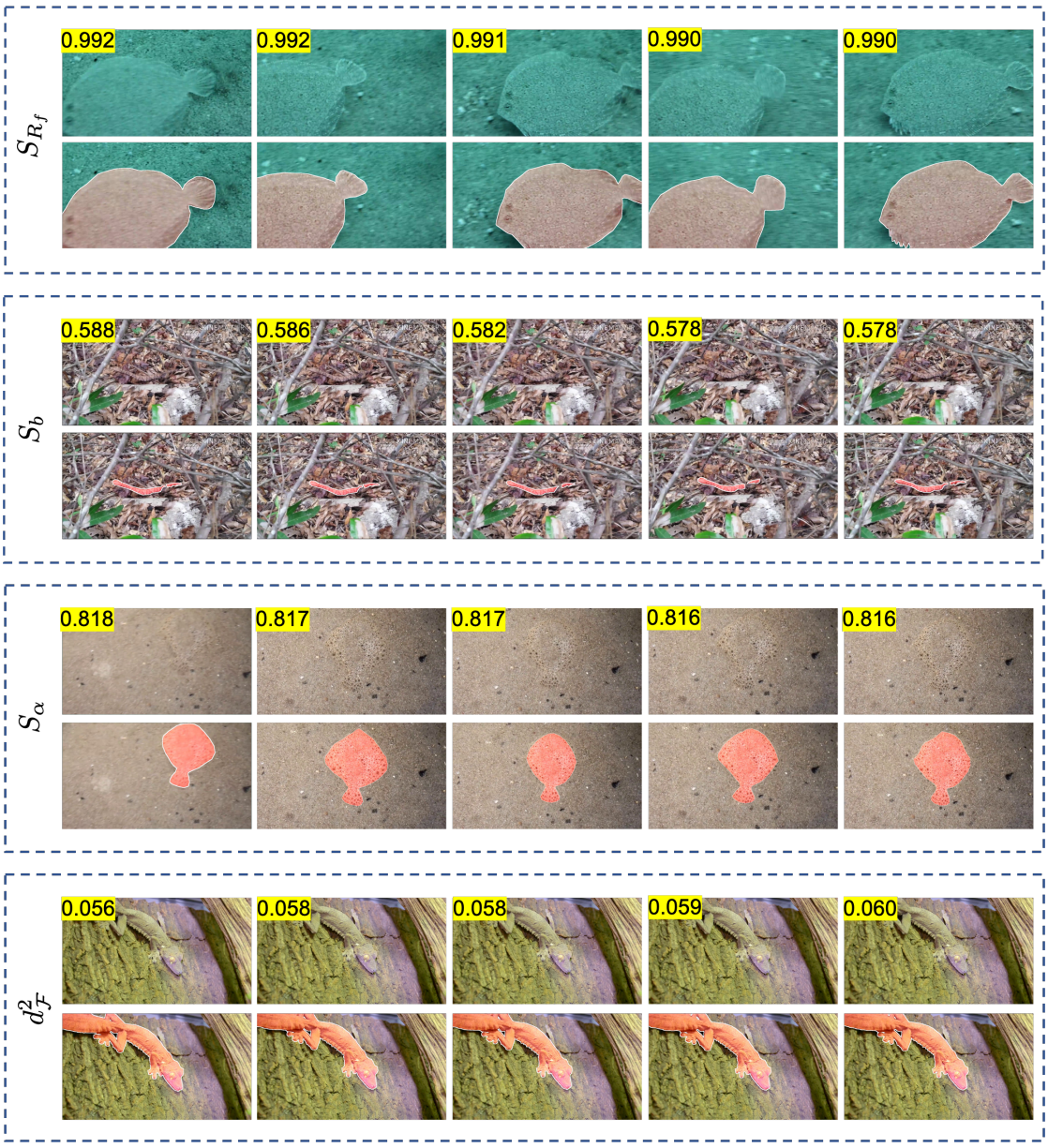


Figure 5.A.3: **Top scored examples from MoCA-Mask.** From top to bottom: S_{Rf}, S_b, S_α and $d^2_{\mathcal{F}}$ rankings. For each frame, we show the corresponding groundtruth mask and the computed score.

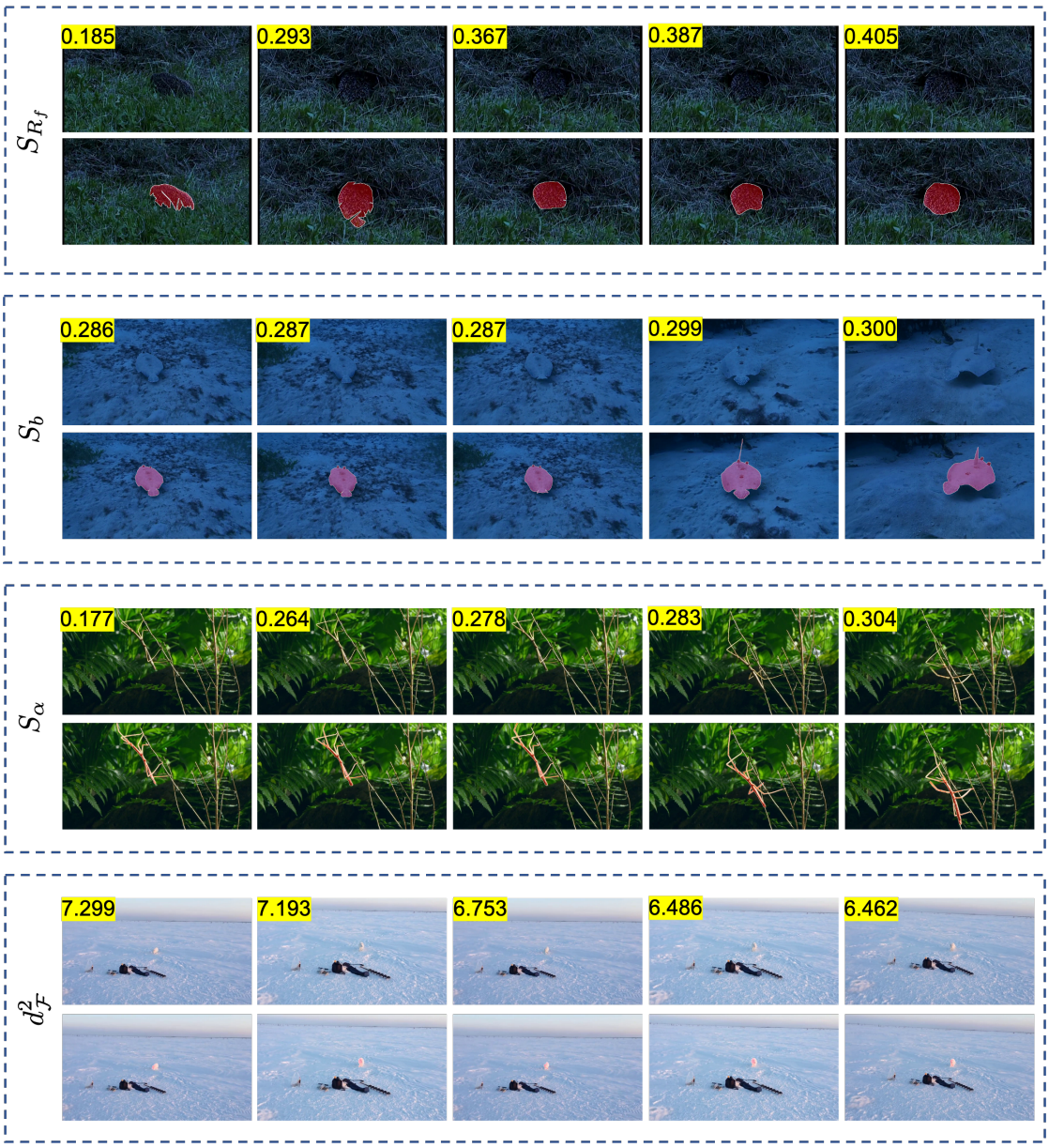


Figure 5.A.4: **Lowest scored examples from MoCA-Mask.** From top to bottom: S_{Rf}, S_b, S_α and $d^2_{\mathcal{F}}$ rankings. For each frame, we show the corresponding groundtruth mask and the computed score.

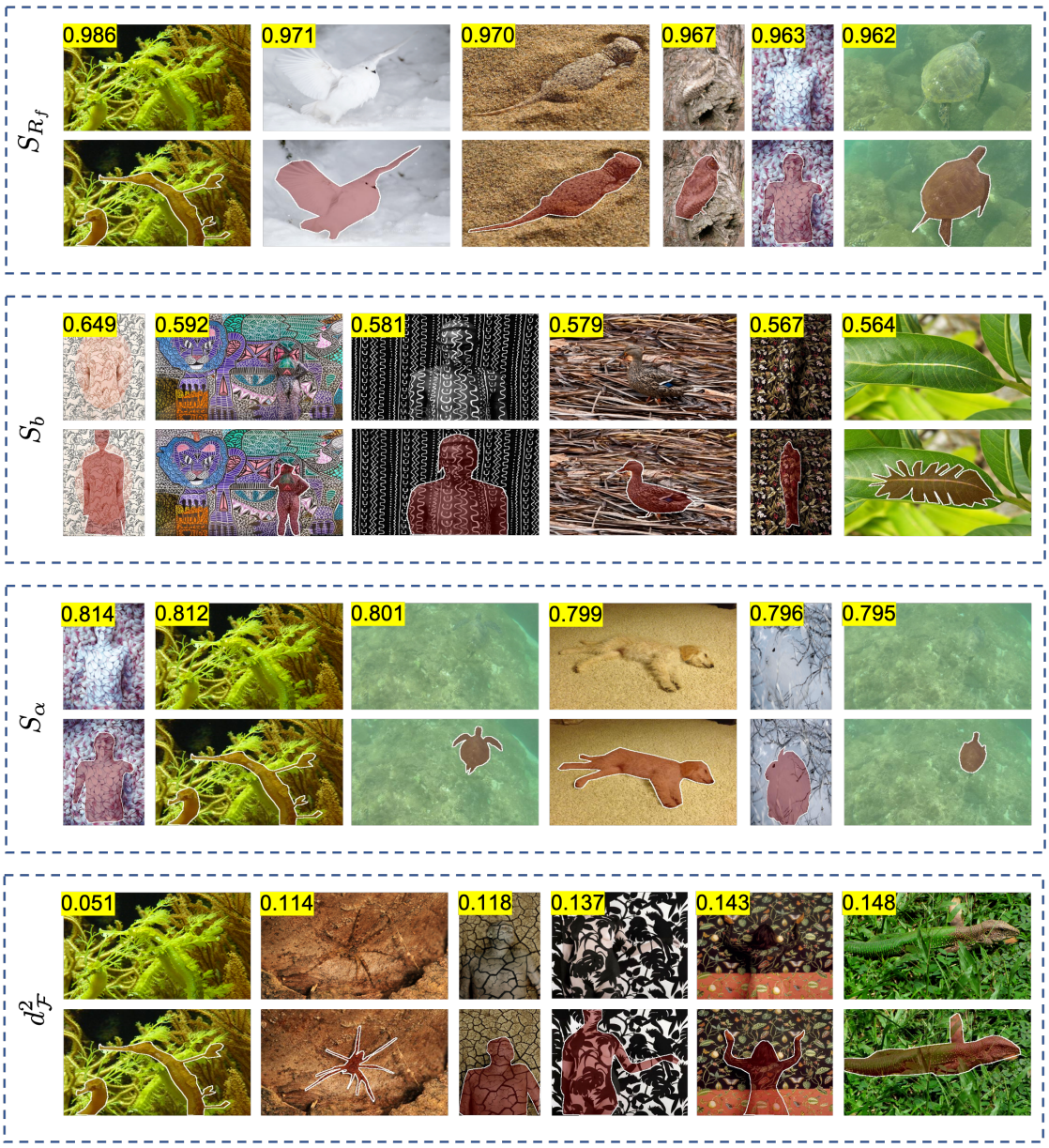


Figure 5.A.5: **Top scored examples from CAMO.** From top to bottom: S_{Rf}, S_b, S_α and $d^2_{\mathcal{F}}$ rankings. For each image, we show the corresponding groundtruth mask and the computed score.



Figure 5.A.6: **Lowest scored examples from CAMO.** From top to bottom: S_{Rf}, S_b, S_α and $d^2_{\mathcal{F}}$ rankings. For each frame, we show the corresponding groundtruth mask and the computed score. While CAMO is a single image dataset, we found multiple images of a red bird with low S_{R_f} and $S_\alpha$scores. Note that these are not part of a sequence.

5.B Synthetic Camouflage Dataset

In this section, we present samples of the synthetically generated camouflage images using our trained generative model and we provide examples of the created video sequences.

5.B.1 Example of generated image samples

We present in Fig 5.B.1 an example of image and mask samples created using our trained generator. Our generator is able to produce realistic object masks of complex shapes and thin structures similar to those usually encountered in the camouflage datasets. We found that the produced images exhibit rich backgrounds with realistic nature-like elements mimicking rocks and coral structures present in the real camouflage data. Finally, when comparing generated samples, we notice a significant improvement of the visual blending of the animal with its background as a result of training the generator with additional intra-image Fréchet loss term.

5.B.2 Example of generated sequences

Fig. 5.B.2 presents examples of generated camouflaged animals sequences. We use translational motion and incorporate static subsequences.

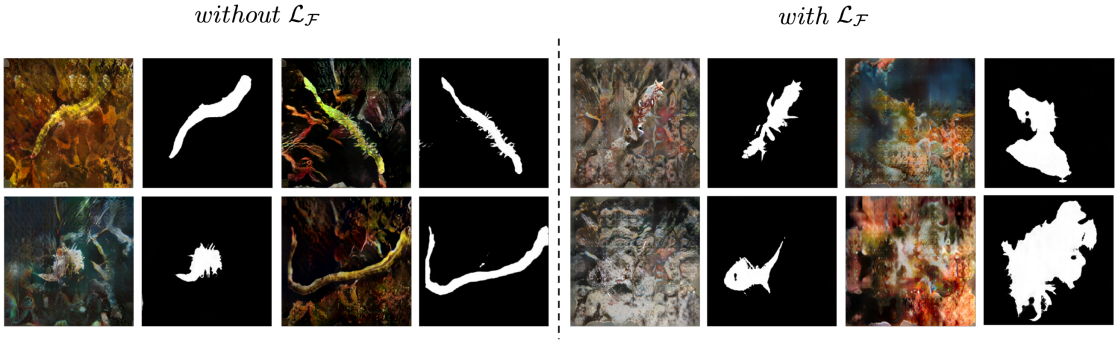


Figure 5.B.1: **Generated camouflage image and mask samples**: For the images on the right, the generator was trained using additional $\mathcal{L}_{\mathcal{F}}$ auxiliary loss.

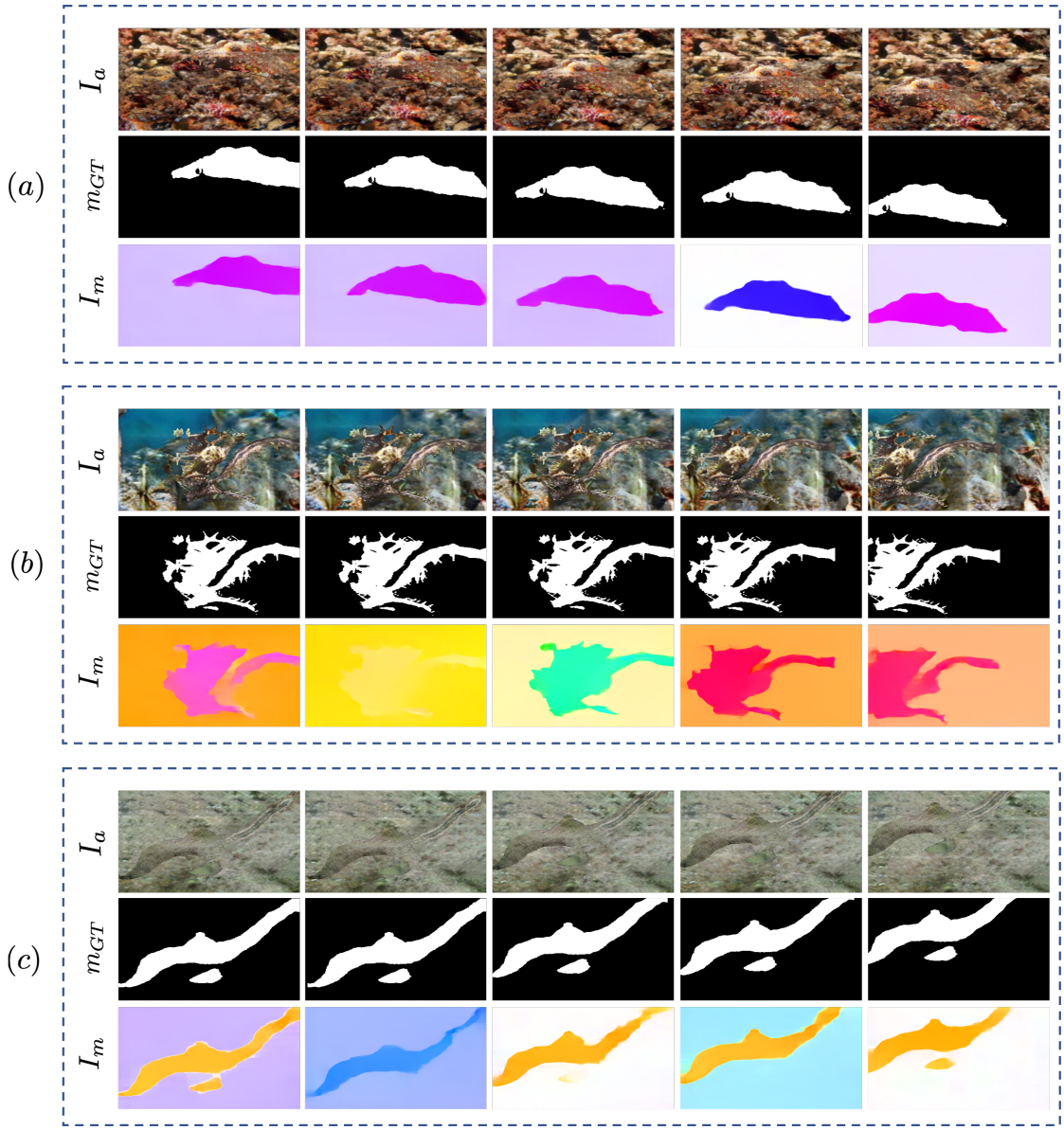


Figure 5.B.2: **Example of generated camouflage sequences.** I_a denotes a sequence of RGB images and I_m the corresponding optical flow sequence.

5.C Moving Camouflaged Animal Segmentation

5.C.1 Model architecture

We detail our motion network architecture in Tab. 5.C.1. We build on the motion segmentation architecture [Lamdouar et al. 2021] that takes a sequence $T = 5$ of optical flow frames and uses spatio-temporal self-attention to learn motion features. The output of the transformer encoder is fed to a pixel decoder with skip connections and further aggregated with motion features via a transformer decoder described in Tab. 5.C.2. The adopted architecture for the appearance encoder, *i.e.* SINetV2 [D. Fan et al. 2021] is described in Tab. 5.C.2. Based on a ResNet50 backbone, this architecture incorporates three levels of resolution: Low-level features ($d = 512$), Mid-level-features ($d = 1024$), and High level features ($d = 2048$). Each feature range is processed by a residual convolutional block. Note that we have omitted the kernel sizes for simplicity. In each residual block, kernel sizes are gradually increased $k \in \{1, 3, 5, 7\}$. To aggregate the appearance and motion features, we use a Transformer Decoder with learnable query features and masked attention [B. Cheng et al. 2022]. The final predicted mask is obtained by multiplying the outputs of linear_2 and conv_2.

	stage	operation	output sizes
	input	–	$T \times 3 \times 256 \times 256$
Motion Encoder	conv1	$[3 \times 3, 64] \times 2$	$T \times 64 \times 256 \times 256$
	mp1	maxpool, stride $= 2$	$T \times 64 \times 128 \times 128$
	conv2	$[3 \times 3, 128] \times 2$	$T \times 128 \times 128 \times 128$
	mp2	maxpool, stride $= 2$	$T \times 128 \times 64 \times 64$
	conv3	$[3 \times 3, 256] \times 2$	$T \times 256 \times 64 \times 64$
	mp3	maxpool, stride $= 2$	$T \times 256 \times 32 \times 32$
	conv4	$[3 \times 3, 512] \times 2$	$T \times 512 \times 32 \times 32$
	mp4	maxpool, stride $= 2$	$T \times 512 \times 16 \times 16$
	conv5	$[3 \times 3, 512] \times 2$	$T \times 512 \times 16 \times 16$
Transformer Encoder	t_{pos}	Embedding $[T \times 512]$	$512 \times T$
	x_{pos}	Embedding $[16 \times 512]$	512×16
	y_{pos}	Embedding $[16 \times 512]$	512×16
	transEnc	input $= 512$, $n_l = 3$ $n_h = 8$, fwd $= 1024$	$(T \times 16 \times 16) \times 512$
Decoder	conv1	$[3 \times 3, 512] \times 2$	$T \times 512 \times 16 \times 16$
	conv^{T}1	$3 \times 3, 256, \text{stride} = 2$	$T \times 256 \times 32 \times 32$
	conv2	$[3 \times 3, 256] \times 2$	$T \times 256 \times 32 \times 32$
	conv^{T}2	$3 \times 3, 128, \text{stride} = 2$	$T \times 128 \times 64 \times 64$
	conv3	$[3 \times 3, 128] \times 2$	$T \times 128 \times 64 \times 64$
	conv^{T}3	$3 \times 3, 64, \text{stride} = 2$	$T \times 64 \times 128 \times 128$
	conv4	$[3 \times 3, 64] \times 2$	$T \times 64 \times 128 \times 128$
	conv^{T}4	$3 \times 3, 64, \text{stride} = 2$	$T \times 64 \times 256 \times 256$

Table 5.C.1: **Architecture of the Motion Network.** In this work, we adopt a ConvNet as motion encoder, and use Transformer Encoder for aggregating the motion temporal information along $T = 5$ frames, followed by ConvNet-based decoder to recover the resolution, similarly to [Lamdouar et al. 2021]. Unless specified otherwise, all convolution layers have stride $= 1$ and padding $= 1$.

	stage	operation	output sizes
	input	–	$T \times 3 \times 256 \times 256$
Appearance Encoder	Backbone	ResNet50	$T \times 512 \times 32 \times 32$ $T \times 1024 \times 16 \times 16$ $T \times 2048 \times 8 \times 8$
	res_block_1	$\left[\text{conv}, 32\right] \times 15$	$T \times 32 \times 32 \times 32$
	res_block_2	$\left[\text{conv}, 32\right] \times 15$	$T \times 32 \times 16 \times 16$
	res_block_3	$\left[\text{conv}, 32\right] \times 15$	$T \times 32 \times 8 \times 8$
	conv_block_1	$\left[\begin{bmatrix}\text{Upsample}\\ \text{conv}, 32\end{bmatrix} \times 4,\ \left[\text{conv}, 64\right] \times 2,\ \left[\text{conv}, 96\right] \times 2,\ \text{conv}, 1\right] \times 1$	$T \times 1 \times 32 \times 32$
	conv_block_2	$\left[\begin{bmatrix}\text{Upsample}\\ \text{conv}, 32\\ \text{conv}, 1\end{bmatrix} \times 3\right] \times 3$	$T \times 4 \times 256 \times 256$
Transformer Decoder	conv_1_1	$[1 \times 1, 512] \times 3$	$T \times 512 \times 256 \times 256$
	transDec	input $= 512$, $n_l = 3$ $n_h = 8$, fwd $= 2048$	512×1
	query_{feat}	Embedding $[1 \times 512]$	512×1
	query_{pos}	Embedding $[1 \times 512]$	512×1
	linear	$[512] \times 2$	512×1
	linear_2	$[256]$	256×1
	conv_2	$[3 \times 3, 256]$	$T \times 256 \times 256 \times 256$

Table 5.C.2: **Architecture of the appearance Network and aggregation.** We adopt a SINetV2 [D. Fan et al. 2021] architecture for encoding appearance. For aggregating the appearance and motion features, we use a Transformer Decoder with learnable query features and masked attention [B. Cheng et al. 2022]. The final predicted mask is obtained by multiplying the outputs of linear_2 and conv_2.

5.C.2 Qualitative results

We show in Fig. 5.C.1 qualitative results of our motion segmentation network. Our model learns motion and appearance cues from the synthetically generated dataset in order to segment the moving camouflaged animal. Note that our model is able to provide accurate segmentations even in the presence of degraded optical flow, *e.g.* the first sequence, or static animal with respect to the background, *e.g.* the momentarily static stick insect depicted in the second sequence and the partially static flower crab spider in the third sequence.

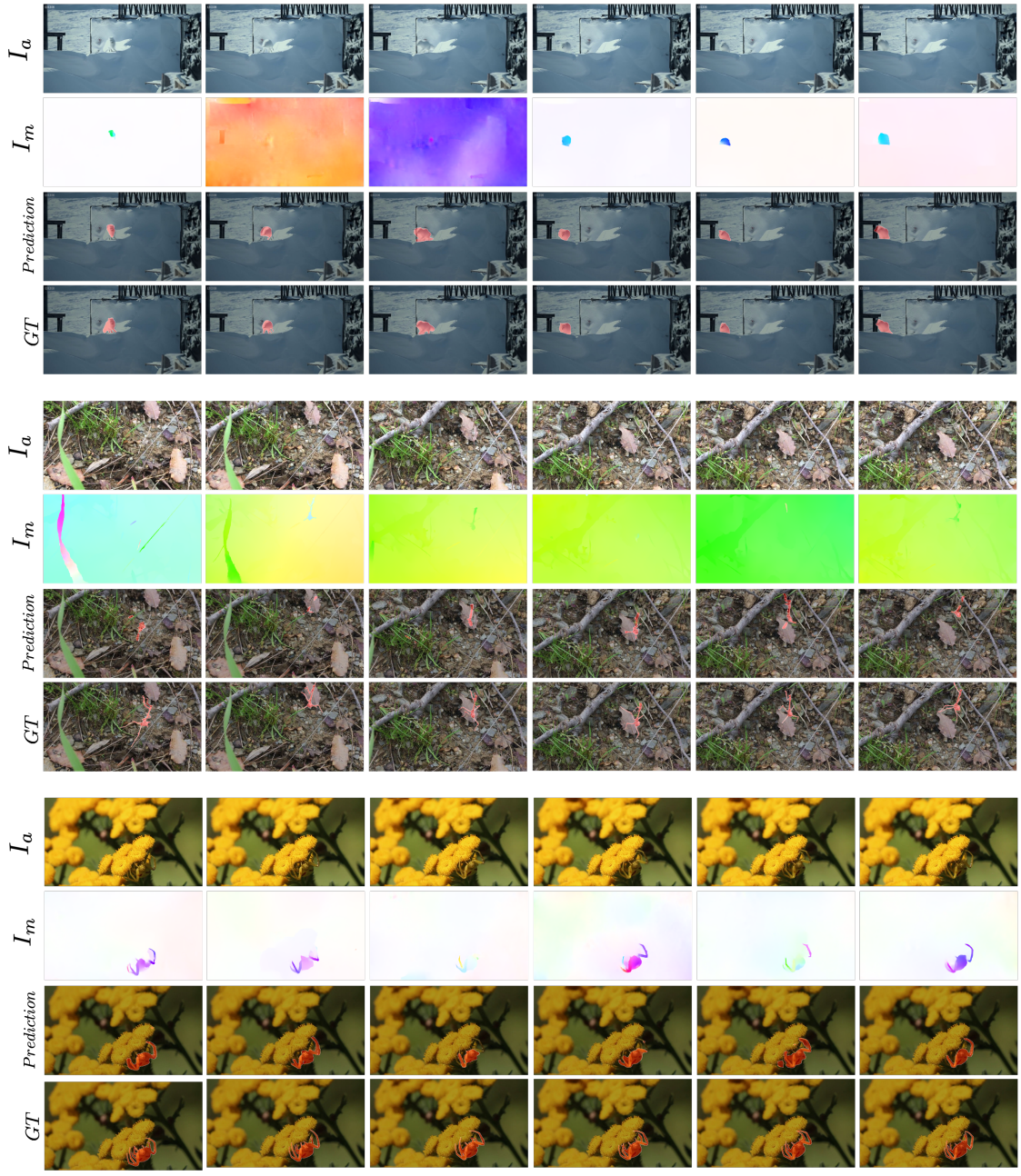


Figure 5.C.1: **Results of our motion segmentation model on sequences from the test subset of MoCA-Mask**. Our model takes as input a sequence of RGB images denoted I_a and a sequence of optical flow I_m and predicts segmentation masks of the moving camouflaged animal.

Chapter 6

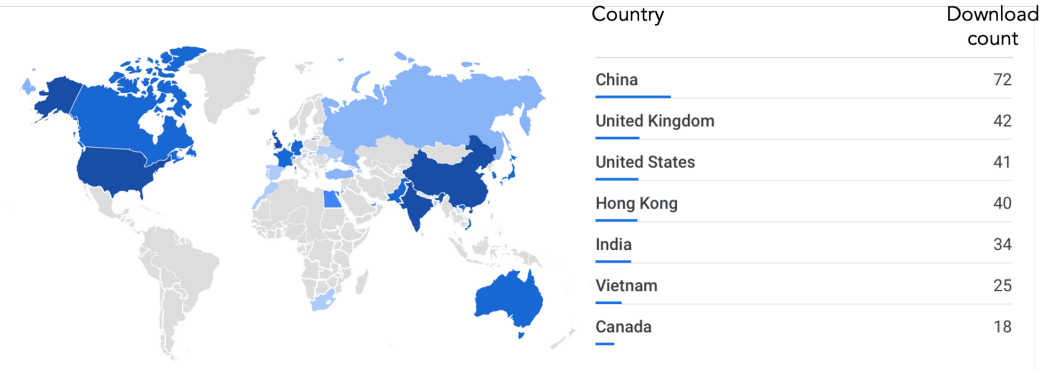
Discussion

In this chapter, we present a summary of the main contributions and impact thus far of the research work conducted in the course of this thesis in Section 6.1 and discuss its ethical implications in Section 6.2. Finally, in Section 6.3, we outline possible directions for future work.

6.1 Achievements and Impact

MoCA : In chapter 2 we presented a dataset for detecting camouflaged animals in videos. With a total of 141 sequences and over 37K frames spanning 67 animal categories, this dataset is to this date the largest for video camouflage breaking. We provide both spatial annotations in the form of bounding boxes, and temporal annotations categorising the animal motion as Locomotion, Deformation and Static. We further provided a curated version, in chapter 4, more suited for motion applications comprising only videos depicting a considerable movement. Since we released our dataset, (see the link https://www.robots.ox.ac.uk/~vgg/data/MoCA/), it has been downloaded over 400 times and publications cited 30 times, for the initial version, and 50 times for the curated version. [X. Cheng et al. 2022] provided a version including segmentation annotations, named MoCA-Mask, with training and test splits, and used it to validate their method for video camouflaged object segmentation. Our dataset has been used among the community as a challenging motion segmentation benchmark [Bideau et al. 2022; J. Xie et al. 2022; Kowal et al. 2022]. [Z. Cheng et al. 2022] used MoCA to validate their motion-based layer-wise separation for unsupervised video frame interpolation, while [F. Y. Zhou

and Ravishankar n.d.] experimented with MoCA for video object detection using spatio-temporal information flow.



Country	Download count
China	72
United Kingdom	42
United States	41
Hong Kong	40
India	34
Vietnam	25
Canada	18

Figure 6.1.1: Geographical distribution of the downloads of MoCA (left) and the most significant download counts per country (right) by January 2023

Video Registration to Enhance Moving Objects In chapter 2, we revisited geometry-based approaches, introducing a background registration method to compensate for the camera motion and enhance the moving object's displacement. We proved that incorporating this step in an end-to-end trainable motion segmentation framework results in a significant improvement in the performance.

A Dual-head Network for Modal and Amodal moving Object Segmentation In chapter 3, we addressed the motion segmentation problem from an object-centric angle and proposed a dual-head architecture capable of achieving modal and amodal perceptions of moving objects.

Synthetic dataset generation for moving object segmentation In chapter 3 we introduced a scalable pipeline for creating synthetic sequences of moving objects. To circumvent the problem of domain gap between the real and synthetic data, we chose to generate the synthetic sequences in the optical flow space. [J. Xie et al. 2022] build on our framework and extend it by incorporating multiple objects and inter-object occlusions. We further extended our method in chapter 5 by developing a synthetic dataset generation pipeline in the RGB space. To close the domain gap that usually occurs in the RGB space, we introduced a method for realistic synthetic camouflage generation by training a convolutional neural network, in an adversarial setting, to output both the image depicting the camouflaged ani-

mal and its corresponding mask. We further introduced a camouflage effectiveness loss to ensure the quality and the challenge of the generated camouflage examples.

Object Discovery via Motion Grouping In chapter 4, we proposed a self-supervised method to segment moving object in videos by learning layered decompositions of optical flow. This publication has been cited 50 times, including [Kipf et al. 2021] who extended our method to multi-objects segmentation on RGB sequences while only using the optical flow as a supervision signal so that at inference time, their method could also be applied to still images where the flow signal is not provided. The abbreviated version of this paper received the Best Paper Award at the *Workshop on Robust Video Scene Understanding: Tracking and Video Segmentation at the Conference on Computer Vision and Pattern Recognition 2021.*

Measuring the effectiveness of camouflage In chapter 5, we introduced the first vision-based method for automatically assessing the challenge of camouflage. We formulated the effectiveness of camouflage using perceptive and probabilistic score functions. We can find useful applications for quantifying the challenge in camouflage examples in ranking the existing datasets by camouflage effectiveness, and computing statistics of the different data split and category subsets. We can also curate these datasets by removing the less challenging examples. Additionally, the investigated perceptual scores can be used to quantify the quality of generated regions for the inpainting task.

6.2 Privacy and Ethical Responsibility

The MoCA dataset contains sequences of camouflaged animals in their natural habitat collected from the YouTube platform. These segments are extracted from documentaries and other shared content of public YouTube channels. While our initial version provided a downloadable archive containing the RGB frames and annotations, in our latest release, we change the download process by providing a list of the urls detailing the starting and ending time of each sequence and the code for download and preprocessing. We believe that this is a simple yet important step towards respecting the ownership of the data. We further propose a contact form enabling the owners of YouTube channels to request the deletion of their

content's url from our benchmark. Note that none of the sequence included in our benchmark contains occurrence of humans and thus it is compliant with the General Data Protection Regulation (EU GDPR).

6.3 Future Work

Extending MoCA Dataset In many cases, camouflaged animals are of highly complex shapes and almost indistinguishable from their background. This adds to the difficulty of annotating an image sequence with static or only partial motion, where the annotator may have to examine a considerable number of previous or next frames in order to perceive the animals' boundaries. For these reasons we will keep our bounding boxes as annotation form. However, we plan to further improve the dataset both quantitatively, by extending to 1000 sequences, and qualitatively, by including motion subcategories *e.g.* walking, flying, crawling, slithering, swimming, running, climbing *etc.* This can be used to further analyse our motion-based detection results as well as other animal activity recognition applications.

Investigating Moving Transparent Objects These highly challenging objects come in a variety of shapes, materials, and properties. While optical flow can provide strong motion cues for some transparent examples such as the rigid glass in Figure 6.3.1, in other cases the challenge is extreme and the optical flow only provides weak information at a few reflection points and potential boundary effects. An example illustrating the latter case can be observed in Figure 6.3.2. To solve such complex cases, a more robust motion representation can be learnt by leveraging contextual information associating different movements in the scene. Indeed, a given video rarely consists of a random collection of motion events but rather depicts causally associated motion patterns. For instance, in Figure 6.3.2, the bubble motion can stand out once we associate it to the juggling movements of the person's hands. Hence, we can distinguish between two main categories of motion:

(i) **Active movements** which involve human and animal action *e.g.* locomotion, pulling, throwing *etc.*

(ii) **Reactive movements** which involve passive objects and things *e.g.* bounc-

ing ball, sliding hockey puck *etc.*

The latter category (ii) represents the inanimate yet movable objects and things of which the motion requires the active movement (i) of an external agent. We can leverage the low-level causality encoded in the optical flow to learn robust action-level representation for our motion classification task. This could be achieved in a self-supervised setting by predicting the future optical flow, similarly to the proposed pipeline by [Han et al. 2019] for RGB images.

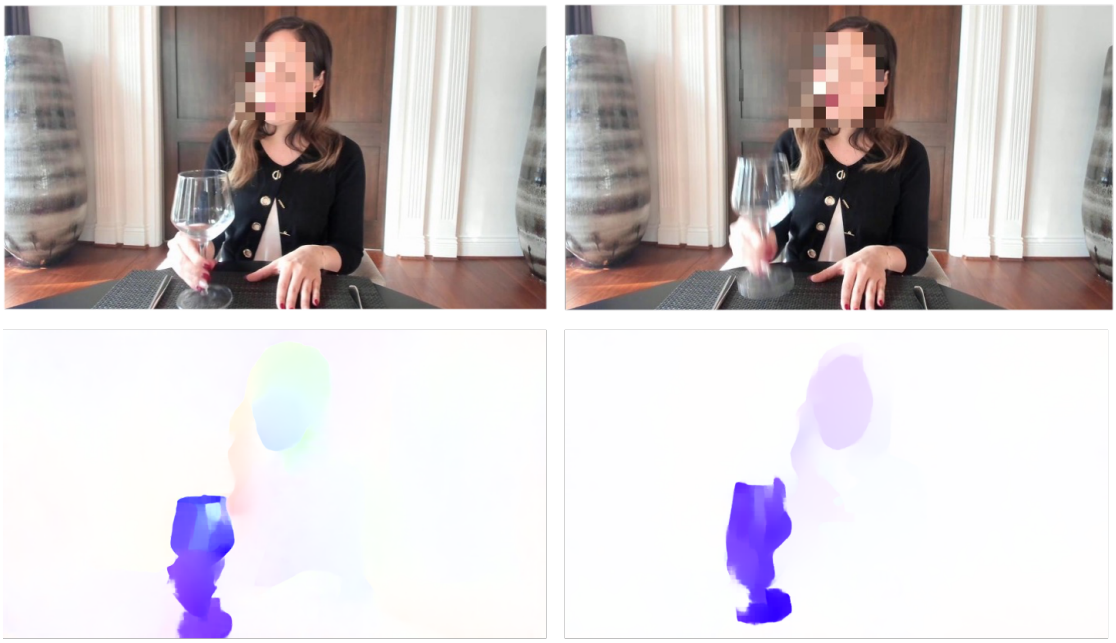


Figure 6.3.1: Examples from the Transparent Object Tracking benchmark (TOTB) of a rigid transparent object (glass). Top: RGB frames, bottom: corresponding optical flows. Note that faces have been anonymised in compliance with the EU General Data Protection Regulation (GDPR).

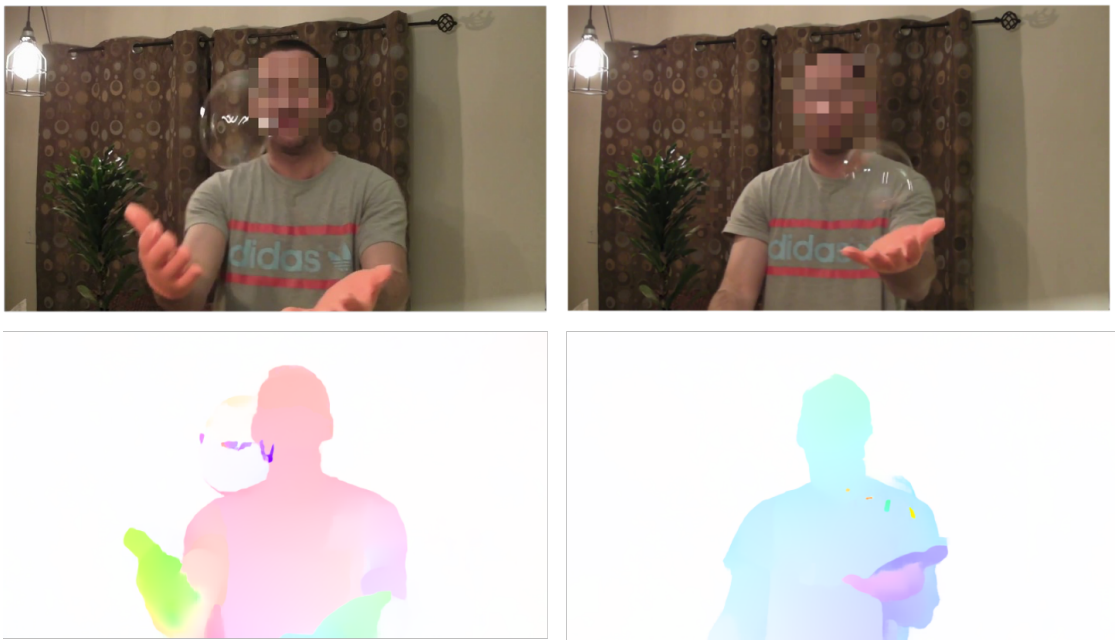


Figure 6.3.2: Examples from the Transparent Object Tracking benchmark (TOTB) of a non-rigid transparent object (soap bubble). Top: RGB frames, bottom: corresponding optical flows. Note that faces have been anonymised in compliance with the EU General Data Protection Regulation (GDPR).

References

Jean-Baptiste Alayrac, João Carreira, Relja Arandjelovic, and Andrew Zisserman (2019a). "Controllable Attention for Structured Layered Video Decomposition". In: *Proc. ICCV*.

Jean-Baptiste Alayrac, João Carreira, and Andrew Zisserman (2019b). "The Visual Centrifuge: Model-Free Layered Video Representations". In: *Proc. CVPR*.

Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski (2011). "A database and evaluation methodology for optical flow". In: *IJCV*.

Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville (2016). "Delving deeper into convolutional networks for learning video representations". In: *Proc. ICLR*.

Olivier Barnich and Marc Van Droogenbroeck (2010). "ViBe: A universal background subtraction algorithm for video sequences". In: *IEEE Transactions on Image processing*.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani (2021). "Is Space-Time Attention All You Need for Video Understanding?" In: *arXiv preprint arXiv:2102.05095*.

Pia Bideau and Erik Learned-Miller (2016a). "A detailed rubric for motion segmentation". In: *arXiv preprint arXiv:1610.10033*.

Pia Bideau and Erik Learned-Miller (2016b). "It's moving! A probabilistic model for causal motion segmentation in moving camera videos". In: *Proc. ECCV*.

Pia Bideau, Erik Learned-Miller, Cordelia Schmid, and Karteek Alahari (2022). "The Right Spin: Learning Object Motion from Rotation-Compensated Flow Fields". In: *arXiv preprint arXiv:2203.00115*.

Pia Bideau, Rakesh R Menon, and Erik Learned-Miller (2018). "MoA-Net: self-supervised motion segmentation". In: *Proc. ECCV Workshop*.

Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother (2017). "DSAC-differentiable RANSAC for camera localization". In: *Proc. CVPR*.

Eric Brachmann and Carsten Rother (2018). "Learning less is more-6d camera localization via 3d surface regression". In: *Proc. CVPR*.

Eric Brachmann and Carsten Rother (2019). "Neural- Guided RANSAC: Learning Where to Sample Model Hypotheses". In: *Proc. ICCV*.

Gabriel J Brostow and Irfan A Essa (1999). "Motion based decompositing of video". In: *Proc. ICCV*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei (2020). "Language Models are Few-Shot Learners". In: *NeurIPS*.

Thomas Brox and Jitendra Malik (2010). "Object segmentation by long term analysis of point trajectories". In: *Proc. ECCV*.

Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner (2019). "MONet: Unsupervised Scene Decomposition and Representation". In: *arXiv preprint arXiv:1901.11390*.

Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black (2012). "A naturalistic open source movie for optical flow evaluation". In: *European conference on computer vision*. Springer.

Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool (2017). "One-shot video object segmentation". In: *Proc. CVPR*.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko (2020). "End-to-end object detection with transformers". In: *Proc. ECCV*.

Jason Chang and John W Fisher (2013). "Topology-constrained layered tracking with latent flow". In: *Proceedings of the IEEE International Conference on Computer Vision*.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever (2020). "Generative Pretraining From Pixels". In: *Proc. ICML*.

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar (2022). "Masked-attention mask transformer for universal image segmentation". In: *Proc. CVPR*.

Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang (2017). "Segflow: Joint learning for video object segmentation and optical flow". In: *Proceedings of the IEEE international conference on computer vision*.

Xuelian Cheng, Huan Xiong, Deng-Ping Fan, Yiran Zhong, Mehrtash Harandi, Tom Drummond, and Zongyuan Ge (2022). "Implicit motion handling for video camouflaged object detection". In: *Proc. CVPR*.

Ziang Cheng, Shihao Jiang, and Hongdong Li (2022). "Unsupervised Video Interpolation by Learning Multilayered 2.5 D Motion Fields". In: *arXiv preprint arXiv:2204.09900*.

Subhabrata Choudhury, Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht (2022). "Guess What Moves: Unsupervised Video and Image Segmentation by Anticipating Motion". In: *Proc. BMVC*.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi (2014). "Describing Textures in the Wild". In: *Proc. CVPR*.

Charles Darwin (1859). "ON THE ORIGIN OF SPECIES." In.

Achal Dave, Pavel Tokmakov, and Deva Ramanan (2019). "Towards segmenting anything that moves". In: *ICCV Workshop on Holistic Video Understanding*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee.

Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich (2016). "Deep image homography estimation". In: *arXiv preprint arXiv:1606.03798*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv preprint arXiv:1810.04805*.

Khoa D Doan, Saurav Manchanda, Fengjiao Wang, Sathiya Keerthi, Avradeep Bhowmik, and Chandan K Reddy (2020). "Image Generation Via Minimizing Fréchet Distance in Discriminator Feature Space." In: *arXiv preprint arXiv:2003.11774*.

Carl Doersch and Andrew Zisserman (2019). "Sim2real transfer learning for 3D human pose estimation: motion to the rescue". In: *Proc. NeurIPS*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. (2021). "An image is worth 16x16 words: Transformers for image recognition at scale". In: *Proc. ICLR*.

Alexei A Efros and Thomas K Leung (1999). "Texture synthesis by non-parametric sampling". In: *Proceedings of the seventh IEEE international conference on computer vision*. IEEE.

Ahmed Elgammal, David Harwood, and Larry Davis (2000). "Non-parametric model for background subtraction". In: *European conference on computer vision*. Springer.

Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner (2020). "GENESIS: Generative scene inference and sampling with object-centric latent representations". In: *Proc. ICLR*.

Mark Everingham, Luc Van Gool, Chris K. I. Williams, John Winn, and Andrew Zisserman (2010). "The PASCAL Visual Object Classes (VOC) Challenge". In: *IJCV*.

Alon Faktor and Michal Irani (2014). "Video Segmentation by Non-Local Consensus voting." In: *Proc. BMVC*.

Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao (2021). "Concealed object detection." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen (2019). "Shifting more attention to video salient object detection". In: *Proc. CVPR*.

Heng Fan, Halady Akhilesha Miththanthaya, Siranjiv Ramana Rajan, Xiaoqiong Liu, Zhilin Zou, Yuewei Lin, Haibin Ling, et al. (2021). "Transparent object tracking benchmark". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Martin A Fischler and Robert C Bolles (1981). "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the ACM*.

Katerina Fragkiadaki, Geng Zhang, and Jianbo Shi (2012). "Video segmentation by tracing discontinuities in a trajectory embedding". In: *Proc. CVPR*.

Maurice Fréchet (1957). "Sur la distance de deux lois de probabilité." In: *Comptes Rendus Hebdomadaires des Seances de L'Academie des Sciences*.

Mario Fritz, Gary Bradski, Sergey Karayev, Trevor Darrell, and Michael Black (2009). "An additive latent feature model for transparent object recognition". In: *Advances in Neural Information Processing Systems*.

Yossi Gandelsman, Assaf Shocher, and Michal Irani (2019). ""Double-DIP": Unsupervised Image Decomposition via Coupled Deep-Image-Priors". In: *Proc. CVPR*.

Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman (2019). "Video Action Transformer Network". In: *Proc. CVPR*.

Melvyn A Goodale and A David Milner (1992). "Separate visual pathways for perception and action". In: *Trends in neurosciences*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). "Generative adversarial nets". In: *NeurIPS*.

Klaus Greff, Raphael Lopez Kaufman, Rishabh Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner (2019). "Multi-object representation learning with iterative variational inference". In: *Proc. ICML*.

Rui Guo, Jasmine Collins, Oscar de Lima, and Andrew Owens (2022). "GANmouflage: 3D Object Nondetection with Texture Fields." In: *arXiv preprint arXiv:2201.07202*.

Tengda Han, Weidi Xie, and Andrew Zisserman (2019). "Video representation learning by dense predictive coding". In: *arXiv preprint arXiv:1909.04656*.

Richard I. Hartley and Andrew Zisserman (2004). *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter (2017). "Gans trained by a two time-scale update rule converge to a local nash equilibrium." In: *NeurIPS*.

Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox (2017). "Flownet 2.0: Evolution of optical flow estimation with deep network". In: *Proc. CVPR*.

Allan Jabri, Andrew Owens, and Alexei A. Efros (2020). "Space-Time Correspondence as a Contrastive Random Walk". In: *Advances in Neural Information Processing Systems*.

Suyog Dutt Jain, Bo Xiong, and Kristen Grauman (2017). "Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos". In: *Proc. CVPR*.

Ge-Peng Ji, Lei Zhu, Mingchen Zhuge, and Keren Fu (2022). "Fast camouflaged object detection via edge-based reversible re-calibration network". In: *Pattern Recognition*.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick (2017). "CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning". In: *Proc. CVPR*.

Nebojsa Jojic, Patrice Simard, Brendan J Frey, and David Heckerman (2001). "Separating appearance from deformation". In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. IEEE.

Yeong Jun Koh and Chang-Su Kim (2017). "Primary object segmentation in videos based on region augmentation and reduction". In: *Proc. CVPR*.

Agastya Kalra, Vage Taamazyan, Supreeth Krishna Rao, Kartik Venkataraman, Ramesh Raskar, and Achuta Kadambi (2020). "Deep polarization cues for transparent object segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Maurice G Kendall (1938). "A new measure of rank correlation". In: *Biometrika*.

Margret Keuper (2017). "Higher-order minimum cost lifted multicuts for motion segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision*.

Margret Keuper, Bjoern Andres, and Thomas Brox (2015). "Motion trajectory segmentation via minimum cost multicuts". In: *Proc. ICCV*.

Kyungnam Kim, Thanarat H Chalidabhongse, David Harwood, and Larry Davis (2004). "Background modeling and subtraction by codebook construction". In: *2004 International Conference on Image Processing, 2004. ICIP'04*. IEEE.

Diederik P. Kingma and Jimmy Ba (2014). "Adam: A Method for Stochastic Optimization". In: *CoRR*.

Thomas Kipf, Gamaleldin F Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff (2021). "Conditional object-centric learning from video". In: *arXiv preprint arXiv:2111.12594*.

Teresa Ko, Stefano Soatto, and Deborah Estrin (2010). "Warping background subtraction". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE.

Takumi Kobayashi and Nobuyuki Otsu (2010). "Von mises-fisher mean shift for clustering on a hypersphere". In: *2010 20th International Conference on Pattern Recognition*. IEEE.

Matthew Kowal, Mennatullah Siam, Md Amirul Islam, Neil DB Bruce, Richard P Wildes, and Konstantinos G Derpanis (2022). "A Deeper Dive Into What Deep Spatiotemporal Networks Encode: Quantifying Static vs. Dynamic Information". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Zihang Lai, Erika Lu, and Weidi Xie (2020). "MAST: A Memory-Augmented Self-Supervised Tracker". In: *Proc. CVPR*.

Zihang Lai and Weidi Xie (2019). "Self-supervised Learning for Video Correspondence Flow". In: *Proc. BMVC*.

Hala Lamdouar, Weidi Xie, and Andrew Zisserman (2021). "Segmenting Invisible Moving Objects". In: *Proc. BMVC*.

Hala Lamdouar, Charig Yang, Weidi Xie, and Andrew Zisserman (2020). "Betrayed by Motion: Camouflaged Object Discovery via Motion Segmentation". In: *Proc. ACCV*.

Trung-Nghia Le, Tam V Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto (2016). "Anabranch network for camouflaged object segmentation". In: *CVIU*.

José Lezama, Karteek Alahari, Josef Sivic, and Ivan Laptev (2011). "Track to the future: Spatio-temporal video segmentation with long-range motion cues". In: *Proc. CVPR*.

Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg (2013). "Video Segmentation by Tracking Many Figure-Ground Segments". In: *Proc. ICCV*.

Ke Li and Jitendra Malik (2016). "Amodal instance segmentation". In: *European Conference on Computer Vision*. Springer.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick (2014). "Microsoft COCO: Common Objects in Context." In.

Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang (2020). "Learning by Analogy: Reliable Supervision from Transformations for Unsupervised Optical Flow Estimation". In: *Proc. CVPR*.

Pengpeng Liu, Michael R. Lyu, Irwin King, and Jia Xu (2019). "SelFlow: Self-Supervised Learning of Optical Flow". In: *Proc. CVPR*.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf (2020). "Object-Centric Learning with Slot Attention". In: *Proc. NeurIPS*.

David G. Lowe (Sept. 1999). "Object recognition from local scale-invariant features". In: *Proc. ICCV*.

Erika Lu, Forrester Cole, Tali Dekel, Weidi Xie, Andrew Zisserman, David Salesin, William T Freeman, and Michael Rubinstein (2020). "Layered Neural Rendering for Retiming People in Video". In: *SIGGRAPH Asia*.

Alan Lukezic, Ziga Trojer, Jiri Matas, and Matej Kristan (2022). "Trans2k: Unlocking the Power of Deep Models for Transparent Object Tracking". In: *arXiv preprint arXiv:2210.03436*.

Yunqiu Lv, Jing Zhang, Yuchao Dai, Aixuan Li, Bowen Liu, Nick Barnes, and Deng-Ping Fan (2021). "Simultaneously localize, segment and rank the camouflaged objects". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Kazuki Maeno, Hajime Nagahara, Atsushi Shimada, and Rin-ichiro Taniguchi (2013). "Light field distortion feature for transparent object recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Sabarinath Mahadevan, Ali Athar, Aljoša Ošep, Sebastian Hennen, Laura Leal-Taixé, and Bastian Leibe (2020). "Making a Case for 3D Convolutions for Object Segmentation in Videos". In: *Proc. BMVC*.

Vijay Mahadevan and Nuno Vasconcelos (2008). "Background subtraction in highly dynamic scenes". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Aravindh Mahendran, James Thewlis, and Andrea Vedaldi (2018). "Cross Pixel Optical-Flow Similarity for Self-Supervised Learning". In: *Proc. ACCV*.

Kevis-Kokitsi Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool (2018). "Video object segmentation without temporal information". In: *IEEE TPAMI*.

Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox (2016). "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation". In: *Proc. CVPR*.

Haiyang Mei, Ge-Peng Ji, Ziqi Wei, Xin Yang, Xiaopeng Wei, and Deng-Ping Fan (2021). "Camouflaged object segmentation with distraction mining". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Anurag Mittal and Dan Huttenlocher (2000). "Scene modeling for wide area surveillance and image synthesis". In: *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*. IEEE.

Anurag Mittal and Nikos Paragios (2004). "Motion-based background subtraction using adaptive kernel density estimation". In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Ieee.

Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua (2018). "Learning to find good correspondences". In: *Proc. CVPR*.

Manjunath Narayana, Allen Hanson, and Erik Learned-Miller (2013). "Coherent motion segmentation in moving camera videos using optical flow orientations". In: *Proceedings of the IEEE International Conference on Computer Vision*.

Andrew Ng, Michael Jordan, and Yair Weiss (2001). "On spectral clustering: Analysis and an algorithm". In: *Advances in neural information processing systems*.

Alaaeldin El-Nouby and Graham W Taylor (2018). "Real-time end-to-end action detection with two-stream networks". In: *arXiv preprint arXiv:1802.08362*.

Peter Ochs and Thomas Brox (2011). "Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions". In: *Proc. ICCV*.

Peter Ochs and Thomas Brox (2012). "Higher order motion models and spectral clustering". In: *Proc. CVPR*.

Peter Ochs, Jitendra Malik, and Thomas Brox (2013). "Segmentation of moving objects by long term video analysis". In: *IEEE transactions on pattern analysis and machine intelligence*.

Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim (2019). "Video object segmentation using space-time memory networks". In: *Proc. ICCV*.

Andrew Owens, Connelly Barnes, Alex Flint, Hanumant Singh, and William Freeman (2014). "Camouflaging an Object from Many Viewpoints." In: *Proc. CVPR*.

Anestis Papazoglou and Vittorio Ferrari (2013). "Fast object segmentation in unconstrained video". In: *Proc. ICCV*.

M Pawan Kumar, Philip HS Torr, and Andrew Zisserman (2008). "Learning layered motion segmentations of video". In: *International Journal of Computer Vision*.

Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung (2016). "A benchmark dataset and evaluation methodology for video object segmentation". In: *Proc. CVPR*.

Ochs Peter, Malik Jitendra, and Brox Thomas (2014). "Segmentation of Moving Objects by Long Term Video Analysis". In: *TPAMI*.

Christina Pieh, Frank Proudlock, and Irene Gottlob (2012). "Smooth pursuit in infants: maturation and the influence of stimulation". In: *British journal of ophthalmology*.

Robert Pless, John Larson, Scott Siebers, and Ben Westover (2003). "Evaluation of local models of dynamic backgrounds". In: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings*. IEEE.

Xavier Soria Poma, Edgar Riba, and Angel Sappa (2020). "Dense extreme inception network: Towards a robust cnn model for edge detection." In: *Proc. WACV*.

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool (2017). "The 2017 davis challenge on video object segmentation". In: *arXiv preprint arXiv:1704.00675*.

Pulak Purkait, Christopher Zach, and Ian Reid (2019). "Seeing Behind Things: Extending Semantic Segmentation to Occluded Regions". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever (2021). "Learning transferable visual models from natural language supervision". In: *arXiv preprint arXiv:2103.00020*.

Hodjat Rahmati, Ralf Dragon, Ole Morten Aamo, Luc van Gool, and Lars Adde (2014). "Motion segmentation with weak labeling priors". In: *German Conference on Pattern Recognition*. Springer.

Vilayanur Subramanian Ramachandran (1985). "Guest editorial: The neurobiology of perception". In: *Perception*.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever (2021). "Zero-Shot Text-to-Image Generation". In: *arXiv preprint arXiv:2102.12092*.

René Ranftl and Vladlen Koltun (2018). "Deep fundamental matrix estimation". In: *Proc. ECCV*.

Anurag Ranjan, Varun Jampani, Lukas Balles, Deqing Sun, Kihwan Kim, Jonas Wulff, and Michael J. Black (2019). "Competitive Collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation". In: *Proc. CVPR*.

Ying Ren, Chin-Seng Chua, and Yeong-Khing Ho (2003). "Statistical background modeling for non-stationary camera". In: *Pattern Recognition Letters*.

Ignacio Rocco, Relja Arandjelovic, and Josef Sivic (2018). "End-to-end weakly-supervised semantic alignment". In: *Proc. CVPR*.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *Proc. MICCAI*. Springer.

Tamar Rott Shaham, Tali Dekel, and Michaeli Tomer (2019). "SinGAN: Learning a Generative Model from a Single Natural Image". In: *Proc. ICCV*.

Yaser Sheikh and Mubarak Shah (2005). "Bayesian modeling of dynamic scenes for object detection". In: *IEEE transactions on pattern analysis and machine intelligence*.

Jianbing Shen, Jianteng Peng, and Ling Shao (2018). "Submodular trajectories for better motion segmentation in videos". In: *IEEE Transactions on Image Processing*.

Feng Shi, Zhong Zhou, Jiangjian Xiao, and Wei Wu (2013). "Robust trajectory clustering for motion segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision*.

Jianbo Shi and Jitendra Malik (2000). "Normalized cuts and image segmentation". In: *IEEE Transactions on pattern analysis and machine intelligence*.

Hedvig Sidenbladh, Michael J Black, and David J Fleet (2000). "Stochastic tracking of 3D human figures using 2D image motion". In: *Proc. ECCV*.

Karen Simonyan and Andrew Zisserman (2014). "Two-stream convolutional networks for action recognition in videos". In: *Advances in neural information processing systems*.

Josef Sivic, Frederik Schaffalitzky, and Andrew Zisserman (2004). "Object Level Grouping for Video Shots". In: *Proc. ECCV*. Springer-Verlag.

Przemysław Skurowski, Hassan Abdulameer, J Błaszczyk, Tomasz Depta, Adam Kornacki, and P Kozieł (2018). "Animal camouflage analysis: Chameleon database." In: *Unpublished manuscript*. https://www.polsl.pl/rau6/chameleon-database-animal-camouflage-analysis/.

Michael Soloveitchik, Tzvi Diskin, Efrat Morin, and Ami Wiesel (2021). "Conditional fréchet inception distance." In: *arXiv preprint arXiv:2103.11521*.

Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam (2018). "Pyramid dilated deeper convlstm for video salient object detection". In: *Proc. ECCV*.

Pratul P. Srinivasan, Richard Tucker, Jonathan T. Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely (2019). "Pushing the Boundaries of View Extrapolation with Multiplane Images". In: *Proc. CVPR*.

Chris Stauffer and W. Eric L. Grimson (2000). "Learning patterns of activity using real-time tracking". In: *IEEE Transactions on pattern analysis and machine intelligence*.

Deqing Sun, Erik B Sudderth, and Michael J Black (2012). "Layered segmentation and optical flow estimation over time". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Deqing Sun, Daniel Vlasic, Charles Herrmann, Varun Jampani, Michael Krainin, Huiwen Chang, Ramin Zabih, William T Freeman, and Ce Liu (2021). "AutoFlow: Learning a Better Training Set for Optical Flow". In: *CVPR*.

Deqing Sun, Jonas Wulff, Erik B Sudderth, Hanspeter Pfister, and Michael J Black (2013). "A fully-connected layered model of foreground and background flow". In: *Proc. CVPR*.

Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz (2018). "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume". In: *Proc. CVPR*.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky (2021). "Resolution-robust Large Mask Inpainting with Fourier Convolutions." In: *arXiv preprint arXiv:2109.07161*.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich (2015). "Going Deeper with Convolutions". In: *Proc. CVPR*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna (2016). "Rethinking the inception architecture for computer vision." In: *Proc. CVPR*.

Rick Szeliski (2004). *Image Alignment and Stitching: A Tutorial*. Tech. rep. URL: https://www.microsoft.com/en-us/research/publication/image-alignment-and-stitching-a-tutorial/.

Matthias Tangemann, Steffen Schneider, Julius Von Kügelgen, Francesco Locatello, Peter Gehler, Thomas Brox, Matthias Kümmerer, Matthias Bethge, and Bernhard Schölkopf (2021). "Unsupervised object learning via common fate". In: *arXiv preprint arXiv:2110.06562*.

Zachary Teed and Jia Deng (2020). "RAFT: Recurrent All-Pairs Field Transforms for Optical Flow". In: *Proc. ECCV*.

Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid (2017a). "Learning motion patterns in videos". In: *Proc. CVPR*.

Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid (2017b). "Learning video object segmentation with visual memory". In: *Proceedings of the IEEE International Conference on Computer Vision*.

Pavel Tokmakov, Cordelia Schmid, and Karteek Alahari (2019). "Learning to segment moving objects". In: *IJCV*.

Philip HS Torr (1998). "Geometric motion segmentation and model selection". In: *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*.

Jolyon Troscianko, John Skelhorn, and Martin Stevens (2017). "Quantifying camouflage: how to predict detectability from appearance". In: *BMC evolutionary biology*.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky (2016). "Instance normalization: The missing ingredient for fast stylization". In: *arXiv preprint arXiv:1607.08022*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). "Attention is All you Need". In: *NIPS*.

Cédric Villani (2009). *Optimal transport: old and new*. Springer.

Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen (2019). "Feelvos: Fast end-to-end embedding learning for video object segmentation". In: *Proc. CVPR*.

Paul Voigtlaender and Bastian Leibe (2017). "Online adaptation of convolutional neural networks for video object segmentation". In: *arXiv*.

Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy (2018). "Tracking Emerges by Colorizing Videos". In: *ECCV*.

Alfred Russel Wallace (1889). "Darwinism: an exposition of the theory of natural selection, with some of its applications". In.

Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu (2019). "Self-Supervised Spatio-Temporal Representation Learning for Videos by Predicting Motion and Appearance Statistics". In: *Proc. CVPR*.

John YA Wang and Edward H Adelson (1994). "Representing moving images with layers". In: *IEEE transactions on image processing*.

Wenguan Wang, Jianbing Shen, Ruigang Yang, and Fatih Porikli (2018). "Saliency-Aware Video Object Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He (2018). "Non-local Neural Networks." In: *Proc. CVPR*.

Max Wertheimer (1923). "Untersuchungen zur Lehre von der Gestalt. II". In: *Psychologische forschung*.

Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland (1997). "Pfinder: Real-time tracking of the human body". In: *IEEE Transactions on pattern analysis and machine intelligence*.

Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim (2018). "Fast video object segmentation by reference-guided mask propagation". In: *Proc. CVPR*.

Jonas Wulff and Michael J. Black (2014). "Modeling Blurred Video with Layers". In: *Proc. ICCV*.

Jonas Wulff and Michael J. Black (2015). "Efficient Sparse-to-Dense Optical Flow Estimation using a Learned Basis and Layers". In: *Proc. ECCV*.

Lu Xiankai, Wang Wenguan, Ma Chao, Shen Jianbing, Shao Ling, and Porikli Fatih (2019). "See More, Know More: Unsupervised Video Object Segmentation With Co-Attention Siamese Networks". In: *Proc. CVPR*.

Christopher Xie, Yu Xiang, Zaid Harchaoui, and Dieter Fox (2019). "Object discovery in videos as foreground motion clustering". In: *Proc. CVPR*.

Enze Xie, Wenjia Wang, Wenhai Wang, Mingyu Ding, Chunhua Shen, and Ping Luo (2020). "Segmenting transparent objects in the wild". In: *European conference on computer vision*. Springer.

Junyu Xie, Weidi Xie, and Andrew Zisserman (2022). "Segmenting Moving Objects via an Object-Centric Layered Representation". In: *NeurIPS*.

Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang (2018). "Youtube-vos: A large-scale video object segmentation benchmark". In: *Proc. ECCV*.

Yichao Xu, Hajime Nagahara, Atsushi Shimada, and Rin-ichiro Taniguchi (2015). "Transcut: Transparent object segmentation from a light-field image". In: *Proceedings of the IEEE International Conference on Computer Vision*.

Yichao Xu, Hajime Nagahara, Atsushi Shimada, and Rin-ichiro Taniguchi (2019). "TransCut2: Transparent object segmentation from a light-field image". In: *IEEE Transactions on Computational Imaging*.

Tianfan Xue, Michael Rubinstein, Ce Liu, and William T. Freeman (2015). "A Computational Approach for Obstruction-Free Photography". In: *ACM Transactions on Graphics (Proc. SIGGRAPH)*.

Jingyu Yan and Marc Pollefeys (2006). "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate". In: *European conference on computer vision*. Springer.

Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie (2021). "Self-supervised Video Object Segmentation by Motion Grouping". In: *Proc. ICCV*.

Gengshan Yang and Deva Ramanan (2018). "Learning to Segment Rigid Motions from Two Frames". In: *Proc. CVPR*.

Yanchao Yang, Brian Lai, and Stefano Soatto (2021). "Dystab: Unsupervised object segmentation via dynamic-static bootstrapping". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yanchao Yang, Antonio Loquercio, Davide Scaramuzza, and Stefano Soatto (2019). "Unsupervised Moving Object Detection via Contextual Information Separation". In: *Proc. ICCV*.

Zhao Yang, Qiang Wang, Luca Bertinetto, Song Bai, Weiming Hu, and Philip H.S. Torr (2019). "Anchor Diffusion for Unsupervised Video Object Segmentation". In: *Proc. ICCV*.

Qiang Zhai, Xin Li, Fan Yang, Chenglizhao Chen, Hong Cheng, and Deng-Ping Fan (2021). "Mutual graph learning for camouflaged object detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo (2022). "Styleswin: Transformer-based gan for high-resolution image generation." In: *Proc. CVPR*.

Felix Yuran Zhou and Roshan Ravishankar (n.d.). "Multiscale Pixel Spatiotemporal Information Flows". In: *A causal view on dynamical systems, NeurIPS 2022 workshop*.

Tianfei Zhou, Shunzhou Wang, Yi Zhou, Yazhou Yao, Jianwu Li, and Ling Shao (2020). "Motion-attentive transition for zero-shot video object segmentation". In: *AAAI*.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely (2018). "Stereo Magnification: Learning View Synthesis using Multiplane Images". In: *SIGGRAPH*.

Zheming Zhou, Xiaotong Chen, and Odest Chadwicke Jenkins (2020). "Lit: Light-field inference of transparency for refractive object localization". In: *IEEE Robotics and Automation Letters*.

Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár (2017). "Semantic amodal segmentation". In: *Proc. CVPR*.

C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski (2004). "High-quality video view interpolation using a layered representation". In: *ACM transactions on graphics (TOG)*.

Appendix A

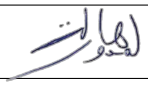
Statement of Authorship

A statement of authorship is provided for each multi-authored paper included in this thesis. The statements describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication, there exists a complete statement that is filled out and signed by the candidate and supervisors.

Statement of Authorship for the paper "Betrayed by Motion: Camouflaged Object Discovery via Motion Segmentation" in Chapter 2.

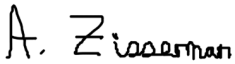
Paper title	Betrayed by Motion: Camouflaged Object Discovery via Motion Segmentation
Authors	**Hala Lamdouar**, Charig Yang, Weidi Xie, Andrew Zisserman
Publication status	Published
Publication details	Proceedings of the Asian Conference on Computer Vision (ACCV), 2020.

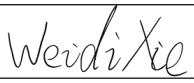
Student Confirmation

Student name	Hala Lamdouar	
Contribution to the paper	• Conception of research ideas • Design and implementation of models • Data collection and preparation • Writing and presentation of the paper	
Signature and Date		January 9th, 2023

Supervisor Confirmation

By signing the Statement of Authorship, the supervisors are certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

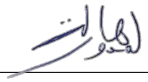
Supervisor name	Prof. Andrew Zisserman	
Supervisor comments		
Signature and Date		January 9th, 2023

Supervisor name	Prof. Weidi Xie	
Supervisor comments		
Signature and Date		January 9th, 2023

Statement of Authorship for the paper "Segmenting Invisible Moving Objects" in Chapter 3.

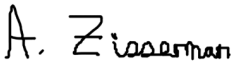
Paper title	Segmenting Invisible Moving Objects
Authors	**Hala Lamdouar**, Weidi Xie, Andrew Zisserman
Publication status	Published
Publication details	Proceedings of the British Machine Vision Conference (BMVC), 2021.

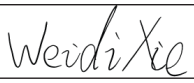
Student Confirmation

Student name	Hala Lamdouar	
Contribution to the paper	• Conception of research ideas • Design and implementation of models • Generation of synthetic datasets • Writing and presentation of the paper	
Signature and Date		January 9th, 2023

Supervisor Confirmation

By signing the Statement of Authorship, the supervisors are certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

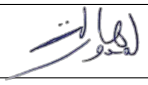
Supervisor name	Prof. Andrew Zisserman	
Supervisor comments		
Signature and Date		January 9th, 2023

Supervisor name	Prof. Weidi Xie	
Supervisor comments		
Signature and Date		January 9th, 2023

Statement of Authorship for the paper "Self-supervised Video Object Segmentation by Motion Grouping" in Chapter 4.

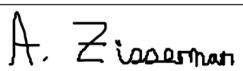
Paper title	Self-supervised Video Object Segmentation by Motion Grouping
Authors	Charig Yang, **Hala Lamdouar**, Erika Lu, Andrew Zisserman, Weidi Xie
Publication status	Published
Publication details	Proceedings of the International Conference on Computer Vision (ICCV), 2021.

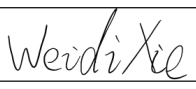
Student Confirmation

Student name	Hala Lamdouar	
Contribution to the paper	• Conception of research ideas • Research of prior work • Implementation and running of some experiments • Writing and presentation of the paper	
Signature and Date		January 9th, 2023

Supervisor Confirmation

By signing the Statement of Authorship, the supervisors are certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

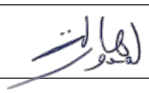
Supervisor name	Prof. Andrew Zisserman	
Supervisor comments		
Signature and Date		January 9th, 2023

Supervisor name	Prof. Weidi Xie	
Supervisor comments		
Signature and Date		January 9th, 2023

Statement of Authorship for the paper "The Making and Breaking of Camouflage" in Chapter 5.

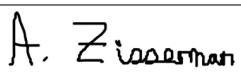
Paper title	The Making and Breaking of Camouflage
Authors	**Hala Lamdouar**, Weidi Xie, Andrew Zisserman
Publication status	Under review

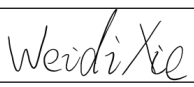
Student Confirmation

Student name	Hala Lamdouar	
Contribution to the paper	• Conception of research ideas • Design and implementation of scoring functions • Generation of synthetic datasets • Writing and presentation of the paper	
Signature and Date		January 9th, 2023

Supervisor Confirmation

By signing the Statement of Authorship, the supervisors are certifying that the candidate made a substantial contribution to the publication, and that the description above is accurate.

Supervisor name	Prof. Andrew Zisserman	
Supervisor comments		
Signature and Date		January 9th, 2023

Supervisor name	Prof. Weidi Xie	
Supervisor comments		
Signature and Date		January 9th, 2023