

Segmenting Invisible Moving Objects - Supplementary material

Hala Lamdouar
lamdouar@robots.ox.ac.uk
Weidi Xie
weidi@robots.ox.ac.uk
Andrew Zisserman
az@robots.ox.ac.uk

Visual Geometry Group
Department of Engineering Science
University of Oxford, UK

Contents

1	Synthetic Dataset Generation	2
1.1	Process of simulation of non-rigid objects	2
1.2	Sequence with static motion	2
1.3	Moving objects passing behind an occluder	3
1.4	Real background motion.	3
1.5	Real shape and background motion.	4
2	Amodal Annotation	5
3	Model Architecture	6
4	Qualitative Results	7
5	Quantitative Results	8

Synthetic Dataset Generation

1.1 Process of simulation of non-rigid objects

Figure 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.

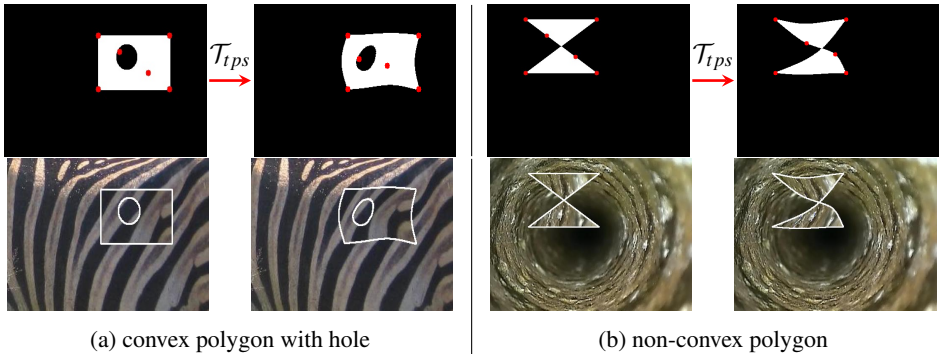


Figure 1: **Example of deformed sprite objects generation.** Generated object masks with the control points used for the thin plate spline \mathcal{T}_{tps} shown in red.

1.2 Sequence with static motion

Figure 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.

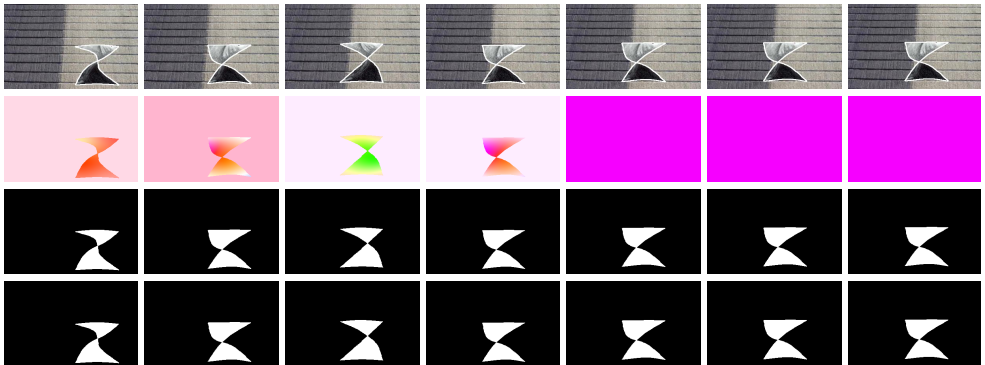


Figure 2: **Example sequence where the moving object becomes static.** From top to bottom: RGB sequence, optical flow, modal segmentations, amodal segmentations.

1.3 Moving objects passing behind an occluder

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

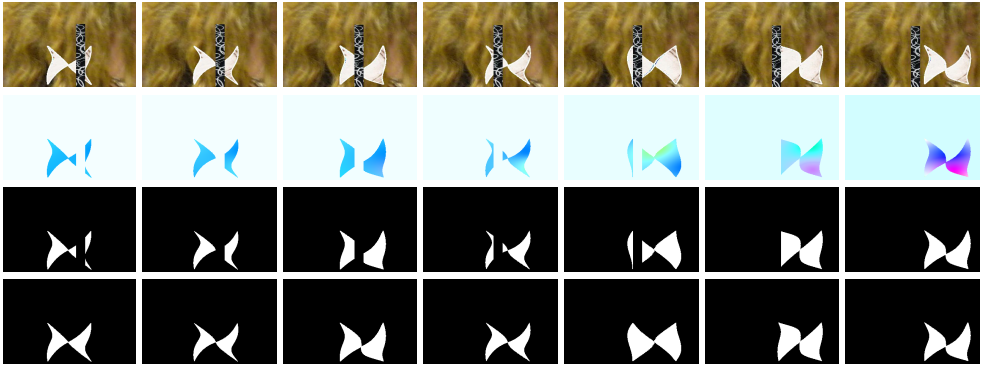


Figure 3: **Example sequence with objects passing behind an occluder.** *From top to bottom:* RGB sequence, optical flow, modal segmentations, amodal segmentations.

1.4 Real background motion.

Figure 4 shows an example of deforming convex polygon undergoing a translational motion, with the presence of a rectangular occluder. Unlike Figures 2 and 3, where the background motion is synthetic, *i.e.* a sequence of translations, Figure 4 presents a real background motion and the optical flow is computed using RAFT [14].

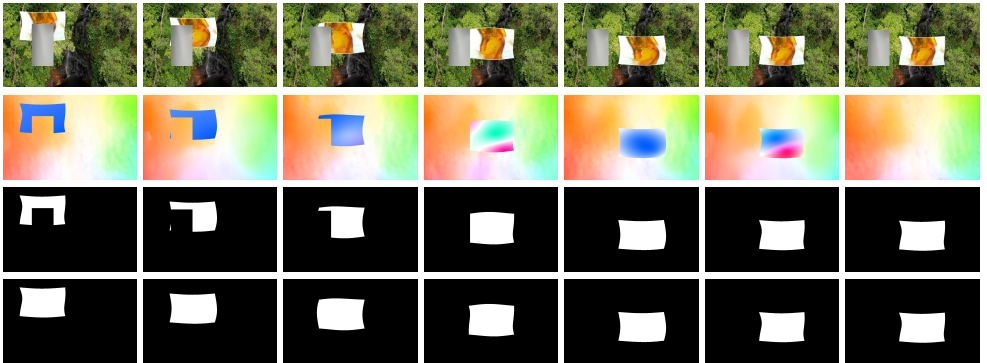


Figure 4: **Example sequence where the background sequence is taken from another video.** *From top to bottom:* RGB sequence, optical flow, modal segmentations, amodal segmentations.

1.5 Real shape and background motion.

Figure 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 and Figure 4, and circular in Figure 5.

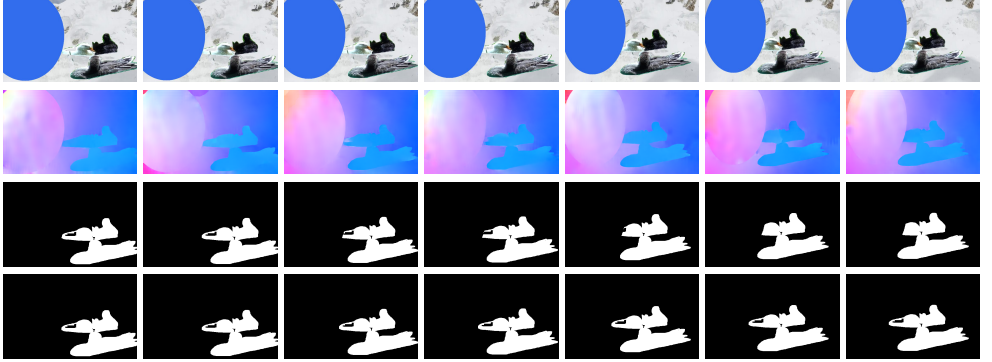


Figure 5: **Example sequence with homographic silhouette and background motion.** *From top to bottom:* RGB sequence, optical flow, modal segmentations, amodal segmentations.

2 Amodal Annotation

We present in this section our amodal annotation process for the validation set of DAVIS 2016. Note that we use these annotations as ground truth for evaluation purposes only. We first use the VIA annotation tool [10] for a polygonal completion of the occluded parts of the original annotation, then generate the completed amodal object mask. We further verify on the RGB space that only occluded regions have been altered and correct the annotation if necessary. Finally, we ensure framewise consistency, as much as is feasible, by comparing framewise annotations.

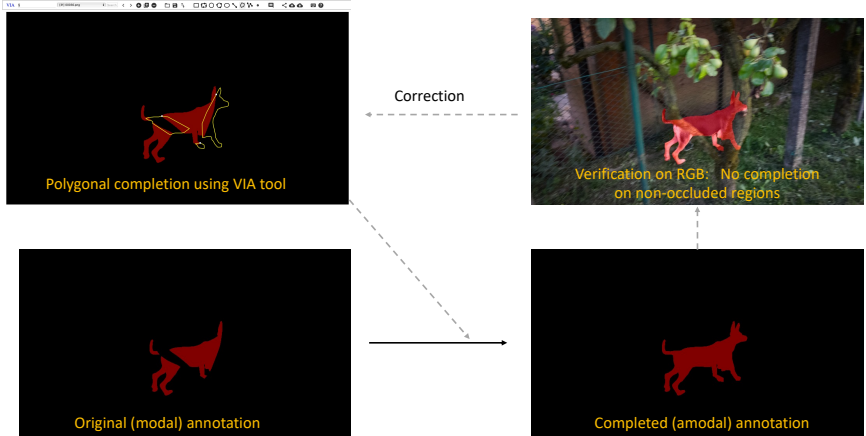


Figure 6: Amodal annotation process.

Figure 7 shows example sequences with the original modal annotations and our amodal annotations.



Figure 7: Example sequences from DAVIS 2016. From top to bottom: RGB sequence, original modal annotation, our amodal annotation.

3 Model Architecture

We present in Table 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 ^T 1	$3 \times 3, 256$, stride = 2	$256 \times 16 \times 16$
	conv2	$[3 \times 3, 256] \times 2$	$256 \times 16 \times 16$
	conv ^T 2	$3 \times 3, 128$, stride = 2	$128 \times 32 \times 32$
	conv3	$[3 \times 3, 128] \times 2$	$128 \times 32 \times 32$
	conv ^T 3	$3 \times 3, 64$, stride = 2	$64 \times 64 \times 64$
	conv4	$[3 \times 3, 64] \times 2$	$64 \times 64 \times 64$
	conv ^T 4	$3 \times 3, 64$, stride = 2	$64 \times 128 \times 128$
	outconv	$7 \times 7, 64$, padding = 3, $7 \times 7, 1$	$1 \times 128 \times 128$

Table 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.

4 Qualitative Results

We present in Figure 8 examples of modal and amodal segmentation results from MoCA and DAVIS 2016 datasets and show sequences from SegTrackV2 in Figure 9.



Figure 8: **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 9: **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.

5 Quantitative Results

We show in this section our per-sequence quantitative results, evaluated using the mean Jaccard measure \mathcal{J} . For MoCA, we adopt the curated version proposed by [14] and use the same evaluation metric, namely localisation rate.

(a) DAVIS2016 dataset		(b) SegTrackv2 dataset	
Sequence	$\mathcal{J}(M)$	Sequence	$\mathcal{J}(M)$
blackswan	0.346	bird of paradise	0.817
bm-x-trees	0.482	birdfall	0.390
breakdance	0.763	bm-x	0.721
camel	0.742	cheetah	0.236
car-roundabout	0.877	drift	0.775
car-shadow	0.868	frog	0.681
cows	0.756	girl	0.820
dance-twirl	0.771	hummingbird	0.557
dog	0.850	monkey	0.752
drift-chicane	0.664	monkeydog	0.198
drift-straight	0.791	parachute	0.913
goat	0.309	penguin	0.591
horsejump-high	0.704	soldier	0.783
kite-surf	0.372	worm	0.449
libby	0.607	Overall	0.620
motocross-jump	0.704		
paragliding-launch	0.618		
parkour	0.760		
scooter-black	0.747		
soapbox	0.821		
Overall	0.678		

Table 2: **Per-sequence results on DAVIS2016 and SegTrackv2.**

Sequence	\mathcal{J}	$\tau_{0.5}$	$\tau_{0.6}$	$\tau_{0.7}$	$\tau_{0.8}$	$\tau_{0.9}$	avg
arabian_horn_viper	76.5	0.969	0.938	0.855	0.855	0.355	0.795
arctic_fox	65.6	0.775	0.650	0.550	0.550	0.400	0.585
arctic_fox_1	95.0	1.000	1.000	1.000	1.000	1.000	1.000
arctic_wolf_0	72.9	0.875	0.834	0.750	0.750	0.448	0.732
arctic_wolf_1	92.2	1.000	1.000	1.000	1.000	1.000	1.000
bear	64.3	0.853	0.660	0.466	0.466	0.148	0.519
black_cat_0	62.0	0.768	0.554	0.375	0.375	0.215	0.458
black_cat_1	8.9	0.073	0.032	0.021	0.021	0.000	0.030
crab	59.7	0.875	0.375	0.125	0.125	0.000	0.300
crab_1	32.4	0.250	0.125	0.000	0.000	0.000	0.075
cuttlefish_0	5.5	0.000	0.000	0.000	0.000	0.000	0.000
cuttlefish_1	10.9	0.000	0.000	0.000	0.000	0.000	0.000
cuttlefish_4	65.0	0.969	0.844	0.157	0.157	0.000	0.426
cuttlefish_5	88.1	1.000	1.000	1.000	1.000	0.813	0.963
dead_leaf_butterfly_1	90.5	1.000	1.000	0.938	0.938	0.938	0.963
desert_fox	35.3	0.150	0.100	0.075	0.075	0.000	0.080
devil_scorpionfish	68.7	0.500	0.500	0.500	0.500	0.500	0.500
devil_scorpionfish_1	92.4	1.000	1.000	1.000	1.000	1.000	1.000
devil_scorpionfish_2	93.8	1.000	1.000	1.000	1.000	1.000	1.000
egyptian_nightjar	77.4	0.989	0.910	0.750	0.750	0.512	0.783
elephant	93.2	1.000	1.000	1.000	1.000	0.938	0.988
flatfish_0	60.3	0.688	0.667	0.646	0.646	0.594	0.649
flatfish_1	67.5	0.862	0.862	0.695	0.695	0.195	0.662
flatfish_2	89.3	1.000	1.000	0.875	0.875	0.834	0.917
flatfish_4	86.2	1.000	1.000	0.989	0.989	0.830	0.962
flounder	71.0	0.782	0.782	0.735	0.735	0.657	0.739
flounder_4	80.3	0.969	0.938	0.782	0.782	0.625	0.820
flounder_5	82.4	0.973	0.917	0.875	0.875	0.625	0.853
flounder_6	74.0	0.834	0.771	0.709	0.709	0.563	0.718
flounder_7	78.8	0.969	0.907	0.797	0.797	0.500	0.794
flounder_8	77.2	0.978	0.864	0.750	0.750	0.591	0.787
flounder_9	24.7	0.188	0.000	0.000	0.000	0.000	0.038
goat_0	68.7	0.823	0.750	0.605	0.605	0.396	0.636
goat_1	76.4	0.969	0.844	0.766	0.766	0.469	0.763
groundhog	62.6	0.750	0.615	0.480	0.480	0.282	0.522
hedgehog_0	57.9	0.650	0.600	0.450	0.450	0.150	0.460
hedgehog_1	62.2	0.875	0.500	0.375	0.375	0.000	0.425
hedgehog_2	90.5	1.000	1.000	1.000	1.000	0.875	0.975
hedgehog_3	43.3	0.407	0.219	0.125	0.125	0.032	0.182
hermit_crab	58.7	0.584	0.542	0.542	0.542	0.334	0.509
ibex	34.0	0.313	0.250	0.188	0.188	0.125	0.213
jerboa	65.3	0.875	0.750	0.438	0.438	0.125	0.526
jerboa_1	51.2	0.500	0.375	0.250	0.250	0.167	0.309
lichen_katydid	46.4	0.344	0.334	0.250	0.250	0.125	0.261
lion_cub_0	84.3	1.000	0.985	0.907	0.907	0.688	0.898
lion_cub_1	51.6	0.615	0.521	0.334	0.334	0.178	0.397
lion_cub_3	18.1	0.209	0.188	0.125	0.125	0.073	0.144
lioness	33.6	0.084	0.000	0.000	0.000	0.000	0.017

marine_iguana	34.6	0.063	0.000	0.000	0.000	0.000	0.013
markhor	82.2	0.917	0.855	0.792	0.792	0.709	0.813
meerkat	95.0	1.000	1.000	1.000	1.000	1.000	1.000
mountain_goat	74.5	1.000	0.959	0.667	0.667	0.250	0.709
nile_monitor_1	64.0	0.771	0.719	0.594	0.594	0.282	0.592
octopus	74.6	0.938	0.855	0.615	0.615	0.375	0.680
octopus_1	38.7	0.365	0.219	0.125	0.125	0.042	0.176
peacock_flounder_0	94.1	1.000	0.988	0.975	0.975	0.963	0.981
peacock_flounder_1	87.0	0.959	0.948	0.917	0.917	0.855	0.920
peacock_flounder_2	90.8	1.000	1.000	1.000	1.000	0.966	0.994
polar_bear_0	78.6	0.907	0.891	0.844	0.844	0.625	0.823
polar_bear_1	58.3	0.697	0.661	0.554	0.554	0.197	0.533
polar_bear_2	83.1	0.938	0.938	0.782	0.782	0.719	0.832
pygmy_seahorse_2	56.2	0.625	0.500	0.396	0.396	0.188	0.421
pygmy_seahorse_4	71.8	1.000	1.000	0.625	0.625	0.042	0.659
rodent_x	70.6	0.938	0.875	0.563	0.563	0.125	0.613
scorpionfish_0	48.1	0.516	0.516	0.500	0.500	0.407	0.488
scorpionfish_1	61.7	0.650	0.600	0.500	0.500	0.450	0.540
scorpionfish_2	84.4	0.903	0.889	0.889	0.889	0.820	0.878
scorpionfish_3	86.1	0.931	0.889	0.778	0.778	0.764	0.828
scorpionfish_4	82.7	1.000	0.938	0.875	0.875	0.594	0.857
scorpionfish_5	92.3	1.000	1.000	1.000	1.000	1.000	1.000
seal_1	94.9	1.000	1.000	1.000	1.000	1.000	1.000
seal_2	77.1	0.834	0.792	0.688	0.688	0.605	0.722
seal_3	66.9	0.875	0.750	0.500	0.500	0.000	0.525
shrimp	90.7	1.000	1.000	1.000	1.000	0.875	0.975
snow_leopard_0	81.5	1.000	0.969	0.907	0.907	0.594	0.876
snow_leopard_1	87.2	1.000	1.000	1.000	1.000	0.875	0.975
snow_leopard_2	92.3	1.000	1.000	1.000	1.000	0.989	0.998
snow_leopard_3	70.9	0.750	0.657	0.563	0.563	0.500	0.607
snow_leopard_6	90.8	1.000	1.000	0.925	0.925	0.875	0.945
snow_leopard_7	66.0	0.875	0.792	0.375	0.375	0.209	0.526
snow_leopard_8	68.3	0.825	0.800	0.550	0.550	0.350	0.615
snowy_owl_0	87.5	0.950	0.950	0.875	0.875	0.825	0.895
spider_tailed_horned_viper_0	42.5	0.500	0.375	0.125	0.125	0.000	0.225
spider_tailed_horned_viper_1	64.7	0.735	0.719	0.641	0.641	0.454	0.638
spider_tailed_horned_viper_2	84.0	0.938	0.938	0.917	0.917	0.855	0.913
spider_tailed_horned_viper_3	83.1	1.000	1.000	0.875	0.875	0.500	0.850
overall	68.6	0.772	0.717	0.623	0.464	0.255	0.566

Table 4: **Per-sequence results on MoCA dataset.** Jaccard measure and successful localization rate for various thresholds.

References

- [1] Abhishek Dutta and Andrew Zisserman. The via annotation software for images, audio and video. In *Proc. ACMM*, volume 27 of *MM 19*, New York, USA, Oct 2019. ACM, ACM. doi: 10.1145/3343031.3350535. to appear in Proceedings of the 27th ACM International Conference on Multimedia (MM 19).
- [2] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proc. ECCV*, 2020.
- [3] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *Proc. ICCV*, 2021.