SAIL-VOS: Semantic Amodal Instance Level Video Object Segmentation – A Synthetic Dataset and Baselines

Yuan-Ting Hu, Hong-Shuo Chen, Kexin Hui, Jia-Bin Huang[†], Alexander Schwing
University of Illinois at Urbana-Champaign [†]Virginia Tech
{ythu2, hschen3, khui3}@illinois.edu, jbhuang@vt.edu, aschwing@illinois.edu

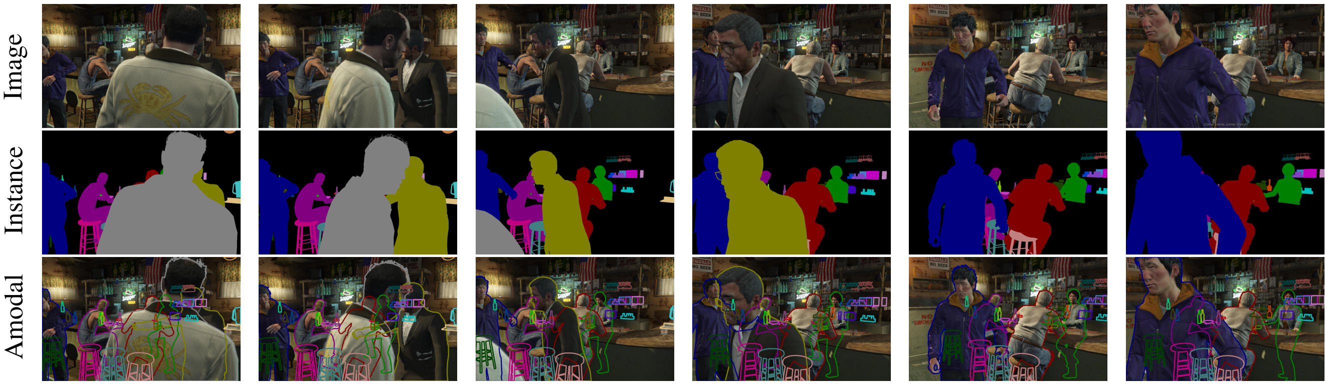

Figure 1: An example sequence of the SAIL-VOS dataset. First row: the original frames. Second row: the instance level segmentation. Third row: the corresponding instance level amodal segmentation.

Abstract

We introduce SAIL-VOS (Semantic Amodal Instance Level Video Object Segmentation), a new dataset aiming to stimulate semantic amodal segmentation research. Humans can effortlessly recognize partially occluded objects and reliably estimate their spatial extent beyond the visible. However, few modern computer vision techniques are capable of reasoning about occluded parts of an object. This is partly due to the fact that very few image datasets and no video dataset exist which permit development of those methods. To address this issue, we present a synthetic dataset extracted from the photo-realistic game GTA-V. Each frame is accompanied with densely annotated, pixel-accurate visible and amodal segmentation masks with semantic labels. More than 1.8M objects are annotated resulting in 100 times more annotations than existing datasets. We demonstrate the challenges of the dataset by quantifying the performance of several baselines. Data and additional material is available at http://sailvos.web.illinois.edu.

1. Introduction

Semantic amodal instance level video object segmentation (SAIL-VOS), *i.e.*, semantically segmenting individual objects in videos even under occlusion, is an important problem for sophisticated occlusion reasoning, depth ordering, and object size prediction. Particularly the temporal sequence provided by a densely and semantically labeled video dataset is increasingly important since it enables assessment of temporal reasoning and evaluation of methods which anticipate the behavior of objects and humans.

Despite these benefits, even for images, amodal segmentation has not been considered until very recently [68, 110, 62, 33, 34, 52, 92, 24, 30]. While the problem is ill-posed, it has been shown that humans are able to predict the occluded regions with high degrees of confidence and consistency [110]. However, the lack of available data makes amodal *image* segmentation a challenging endeavor, even today. Concretely, no dataset was available until Maire *et al*. [68] released 100 meticulously annotated images in 2013. Zhu *et al*. [110] followed up in 2015 by annotating 5000 images. In 2018 two more datasets were released by Ehsani *et al*. [24] and Follman *et al*. [29].

Unfortunately, the situation is worse when considering videos. While video object segmentation data, *e.g.*, SegtrackV1 [97], Youtube-Objects [80], FBMS [71], JumpCut [28], DAVIS16 [76], and instance level video object segmentation data, *e.g.*, SegtrackV2 [61], DAVIS17 [79], Youtube-VOS [101], is available these days, no dataset permits direct training of semantic amodal instance level *video* object segmentation (SAIL-VOS). This isn't surprising when considering that it is time-consuming and expensive to collect such a dataset. Nevertheless, its use for the aforementioned applications (temporal reasoning, anticipating behavior, depth ordering, *etc*.) is immediately obvious.

To improve the situation, we make a first step in the direction of SAIL-VOS, presenting a new dataset and baselines by leveraging Grand Theft Auto V (GTA-V) as a reasonably realistic environment simulator. The collected dataset improves upon existing amodel image segmentation data in multiple ways: (1) the data is semantically labeled at an object instance level based on COCO's categories but not including all classes defined in COCO; (2) in addition to classical modal segmentation, an amodal instance level segmentation is made available; (3) the proposed dataset consists of densely sampled and completely annotated video frames, permitting models to leverage dense temporal information; (4) diverse amodal 3D pose information for humans is available as well. The latter can be used as an additional cue or can be used as training data for 3D pose detectors.

Even though it isn't real data just yet, we believe that usage of data from an environment simulator such as GTA-V is an important first step to assess suitability of models in the absence of real-world data and to gauge the use of collecting real-world data. Moreover, at least for images, the proposed dataset provides an opportunity to assess transfer-learning abilities by leveraging the publicly available amodal image data provided by Zhu *et al*. [110]. We therefore hope the collected data will stimulate research in two directions: (1) methods and techniques for SAIL-VOS, a task that is significantly more challenging than classical instance level video object segmentation; (2) transfer-learning techniques.

The proposed error metrics for evaluation on the SAIL-VOS dataset address both directions. Beyond introducing the dataset we also provide baseline methods to assess and demonstrate the challenges of SAIL-VOS.

2. Related Work

We now discuss directions related to SAIL-VOS.

Image Segmentation research can be traced back to the 1970s [72] and gained increasingly more attention once a focus on objects and semantics was established in the late 1980s [23]. Early segmentation methods group perceptually similar pixels, *e.g*., color image segmentation by Comaniciu and Meer [19], normalized cuts by Shi and Malik [88] or implicit shape models [60]. Popular datasets for segmentation include the BSDS dataset [4]. Classifiers for semantic segmentation were considered by Konishi and Yuille [57] demonstrating results on the Sowerby Image Database and the San Francisco Database. As early as 2004, He *et al*. [40] applied Conditional Random Fields (CRFs) [59], segmenting a 100-image subset of the Corel image database into 7 classes, and segmenting the Sowerby Image Database (104 images) into 8 classes. Semantic segmentation was further popularized by combining Random Forests with CRFs as proposed by Shotton *et al*. [90] who used 591 images containing 21 classes. This latter dataset was a precursor to the Pascal VOC 2007 segmentation taster [25] which was significantly extended in 2012, also via the Semantic Boundaries Dataset [36]. Deep net based methods have recently been predominant, starting with [32, 86, 67, 13, 109]. Recent developments include atrous/dilated spatial pyramid pooling and encoders [14] which achieved state-of-the-art on the Pascal VOC 2012 dataset at the time of writing. Many more datasets for specific domains such as autonomous driving have recently been introduced as well and we highlight them below.

Semantic Instance-level Image Segmentation provides a more detailed decomposition of the scene into individual instances, which is, among others, useful for count-based search, differentiation of multiple objects, *etc*. Some techniques infer depth-ordering [33, 103, 94, 108], while others are based on a combination of detection and segmentation [104, 37, 38], and yet others combine several oversegmentations [93]. Reference images are used in [39] and the approach is evaluated on the TUD Pedestrian [3] and the Polo dataset [107]. Another common technique is partitioning of an initial semantic segmentation [56, 5, 6, 66]. A variety of other techniques based on voting, refinement, multi-task learning and grouping, *e.g*., [8, 84, 55, 99, 77, 78, 21, 22, 64], have been investigated.

Frequently used recent datasets for instance-level image segmentation are Pascal VOC 2012 [25], NYUv2 [91], MS COCO [65], CityScapes [20], Berkeley Deep Drive [106], KITTI [2] and Mapillary Vistas [70].

Amodal Image/Instance Segmentation, despite obvious ambiguities, was shown to be a task where different operators agree to a reasonable degree on the annotation [110]. To facilitate the task Maire *et al*. [68] meticulously annotated 100 images and Zhu *et al*. [110] provided 5000 COCO annotations. Early object agnostic methods are based on contour completion [33, 34, 52, 92].

Explicit occlusion reasoning [43, 31, 74, 16] is related in that occluded objects are detected using bounding boxes or occlusion of specific objects are modeled, *e.g*., people.

Amodal instance segmentation without amodal data was considered by Li and Malik [62]. It is proposed to sidestep the lack of training data by synthetically adding occlusion and retaining the original semantic segmentation mask. The authors found this technique to be quite robust. However, occlusion patterns are random in this case which is unlikely for real-world data.

Very recently a generative adversarial net based method and the new synthetic indoor scene dataset 'DYCE' [24] comprised of five living rooms and six kitchens has been released. Our SAIL-VOS dataset differs in that the scenes are from both indoor and outdoor settings and consequently

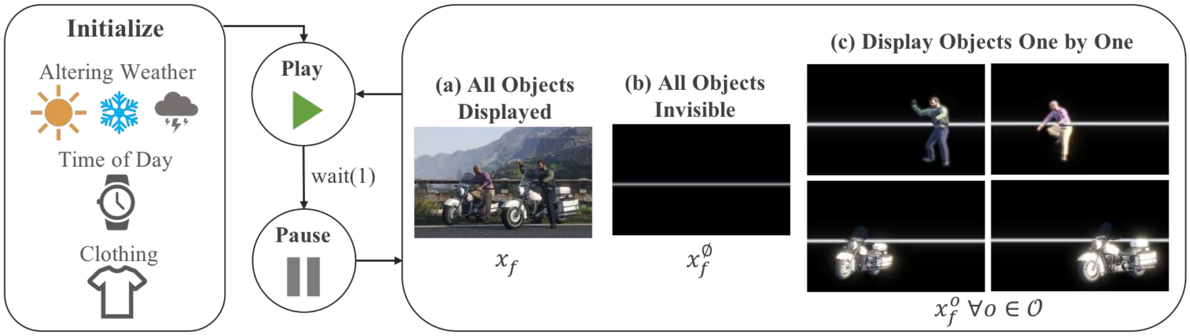
Initialize
Altering Weather
Time of Day
Clothing
Play
wait(1)
Pause
(a) All Objects Displayed
x_f
(b) All Objects Invisible
$x_f^\emptyset$
(c) Display Objects One by One
$x_f^o \ \forall o \in \mathcal{O}$

Figure 2: The dataset collection pipeline. For every sequence, we randomly initialize the weather, the time of day and the clothing of appearing characters. After initialization, we play the scene, wait for 1 milliseconds (ms) and pause to collect the frames, including (a) the original frame with all objects displayed, (b) the empty background with all objects invisible and (c) screenshots with one object displayed at a time. We then resume the game, pause after 1ms and collect data again.

much richer (see Fig. 1). Moreover, we consider video data.

Even more recently the 'Densely Segmented Supermarket' (D2S) dataset [29] has been presented, focusing on a warehouse setting, *i.e.*, groceries from a top-down view. Our proposed dataset is again much more diverse.

Video Object Segmentation data annotation is much more time-consuming than labeling of images. It is therefore not surprising that first datasets, which focus on a single object per video, didn't appear until recently [97, 80, 71, 28, 76]. A variety of methods have tackled the unsupervised (*i.e.*, no object annotation available during testing) [73, 71, 27, 44, 95, 96, 48] and the semi-supervised (*i.e.*, first frame annotation available during testing) [12, 54, 49, 11, 53, 98, 45, 18, 63, 105, 17, 15, 102, 100] setting.

Instance-level Video Object Segmentation data has been made available even more recently [61, 79, 101] since it requires annotation of multiple objects per video. A variety of techniques have also been proposed, *e.g.*, [85, 45, 46]. To the best of our knowledge, no data is publicly available for amodal instance-level video object segmentation.

We note that particularly the autonomous driving based semantic instance level segmentation data is related to video segmentation as frames are extracted from driving vehicles.

Synthetic Data has been used for many tasks in the computer vision community for a long time, particularly if real data is hard to obtain. For instance, synthetic data has been used for optical flow [42, 41, 9, 7, 69, 81, 58], robustness estimation of features [51], visual odometry [35, 81], hand tracking [87], shape from shading [82], semantic segmentation [83, 50, 81, 58], amodal segmentation [52, 24], multi-view stereopsis [47] and human pose estimation [89, 26].

This work differs from the aforementioned ones in that we collect a dataset for the task of SAIL-VOS using the GTA-V environment simulator. To obtain instance level labels our data collection procedure differs significantly from any of the aforementioned works. We think this data will help our community assess the use cases of SAIL-VOS data, and we hope it will inspire research in a new direction. Beyond presenting the dataset we also propose a few simple baseline methods and illustrate arising challenges that future models have to meet.

3. Data Collection

We use the game GTA-V to collect the SAIL-VOS dataset. In order to compute the amodal segmentation, we need to get information from the *occluded region*, which is not visible on the screen. We therefore interact with the game engine and successively toggle the visibility of objects such that we can perceive each object as a whole.

We illustrate our dataset collection pipeline in Fig. 2. Specifically, we play the scenes to collect video sequences and their annotations. For each sequence, we randomly alter the weather condition, the time of day and the clothing of characters which appear in the video sequence before playing the video. After waiting for 1 millisecond we pause the game and collect the data for this particular frame. Due to a delay between sending a pause signal and the game reaction, we obtain a sampling rate of around 8 frames per second. We repeat the process by resuming the video, waiting for another 1 milliseconds before recording the next frame. For every frame we pause at, we record screenshots turning the visibility of all objects once on (Fig. 2 (a)) and once off (Fig. 2 (b)). In addition, we toggle the visibility of all objects, displaying them one by one at a time and acquire screenshots as shown in Fig. 2 (c). We use the Script Hook V library [10] for altering the weather, the time of day, the clothing and pausing the game as well as toggling the visibility of objects. Along with the screenshots (RGB images), we hook into DirectX functions to access the GPU resources and save the corresponding stencil and depth buffers as shown in Fig. 3. The DirectX hooks are based on the GameHook library [58]. The stencil buffer contains the semantic label of each pixel at the class level, instead of at the instance level, *e.g.*, all pixels belonging to a person will be

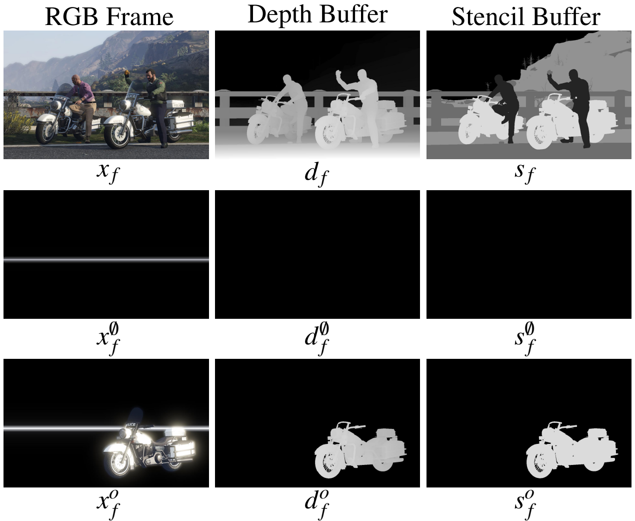

Figure 3: Along with every recorded image, we also record the depth buffer (the second column) and the stencil buffer (the third column). We collect data with all objects displayed (the first row), all object invisible (the second row), and an example where there is only one object displayed at a time (the third row).

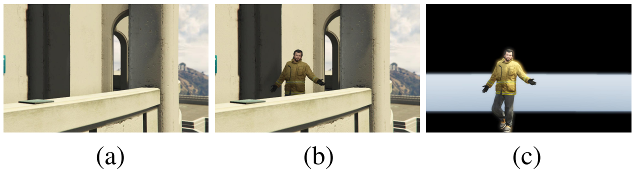

Figure 4: There are objects for which visibility cannot be toggled using Script Hook V. (a) The frame with objects in $\mathcal{O}$ not displayed. The visibility of the building cannot be toggled by Script Hook V. (b) x_f^o without a clean background. Note that we cannot compute the amodal segmentation as the object is partially occluded. (c) x_f^o with a clean background. A clean background is obtained by hooking into DirectX draw functions.

assigned the same value no matter whether they belong to the same person or not.

Amodal segmentation: To describe computation of the amodal segmentation formally we introduce some notation. We collect an RGB image x_f, the depth buffer d_f and the stencil buffer s_f for each frame $f \in \{1, \ldots, F\}$ as shown in the first row of Fig. 3. For each frame f we also capture the image, depth and stencil buffer $x_f^{\emptyset}$, $d_f^{\emptyset}$ and $s_f^{\emptyset}$ which do not show any objects (see Fig. 3, second row). We subsume all objects of frame f for which we can toggle visibility in the set $\mathcal{O}_f$. We obtain this set via Script Hook V. For each object $o \in \mathcal{O}_f$, we also capture the RGB image x_f^o, the depth buffer d_f^o, and the stencil buffer s_f^o showing *only one object* o (see third row of Fig. 3). Note that visible objects exist for which we cannot toggle the visibility using Script Hook V, such as the building shown in Fig. 4 (a). Obviously we cannot compute the amodal segmentation of an object if it is occluded by other objects for which we cannot toggle visibility, as shown in Fig. 4 (a, b). To address this issue we hook into the DirectX draw functions when recording x_f^o, $x_f^{\emptyset}$ and their corresponding depth and stencil buffer. Specifically, we obtain a clean scene by not rendering any objects but the one currently under consideration. Hence the hook function only issues a rendering call for the currently targeted object and ignores all other rendering requests.

To compute the amodal mask a_f^o of object $o \in \mathcal{O}$, we fuse the information of the depth buffer and the stencil buffer, instead of using purely the depth buffer or the stencil buffer. We found using the combination of both to be important for higher accuracy. For instance, we found that the depth of an object might be slightly altered after toggling the visibility, especially for soft objects such as clothes. This randomness during rendering, presumably added to increase game realism, results in artifacts when computing the segmentation exclusively based on the depth mask. Using purely depth information is therefore not enough, while the stencil buffer contains only class-level semantic segmentation instead of instance-level segmentation. To obtain accurate segmentations, we therefore first compute the amodal segmentation based on depth, $a_{f,d}^o$, by comparing d_f^o with $d_f^{\emptyset}$, *i.e.*,

$$a_{f,d}^o = \delta(d_f^o \neq d_f^{\emptyset}).$$

Here, δ returns a binary mask indicating which pixels of the object depth map differ from the background depth map. Similarly, we compute the amodal mask $a_{f,s}^o$ based on stencil information using the above equation but replacing the depth buffer with the stencil buffer.

We then fuse $a_{f,d}^o$ and $a_{f,s}^o$ to get the amodal mask a_f^o via

$$a_f^o = a_{f,d}^o \oplus a_{f,s}^o,$$

where '$\oplus$' denotes a logical OR operation to combine the amodal segmentation masks.

To compute the visible segmentation, we also first compute the depth-based modal mask $m_{f,d}^0$ via

$$m_{f,d}^o = \delta(d_f^o = d_f) \cdot \delta(a_f^o = 1).$$

We also compute the visible mask $m_{f,s}^o$ using the above equation while replacing the depth buffer with the stencil buffer. To obtain the visible segmentation candidate m_f^o we fuse $m_{f,d}^o$ and $m_{f,s}^o$ via

$$m_f^o = m_{f,d}^o \cdot m_{f,s}^o.$$

Object tracking: Every object in $\mathcal{O}$ is assigned a unique ID in the game and we can get the IDs by accessing the rendering engine using the Script Hook library. As the IDs do not change across frames, we are able to keep track of objects via their IDs.

Semantic class label: Manually, we also assign a class label to each object, defining a total of 162 classes. To assign

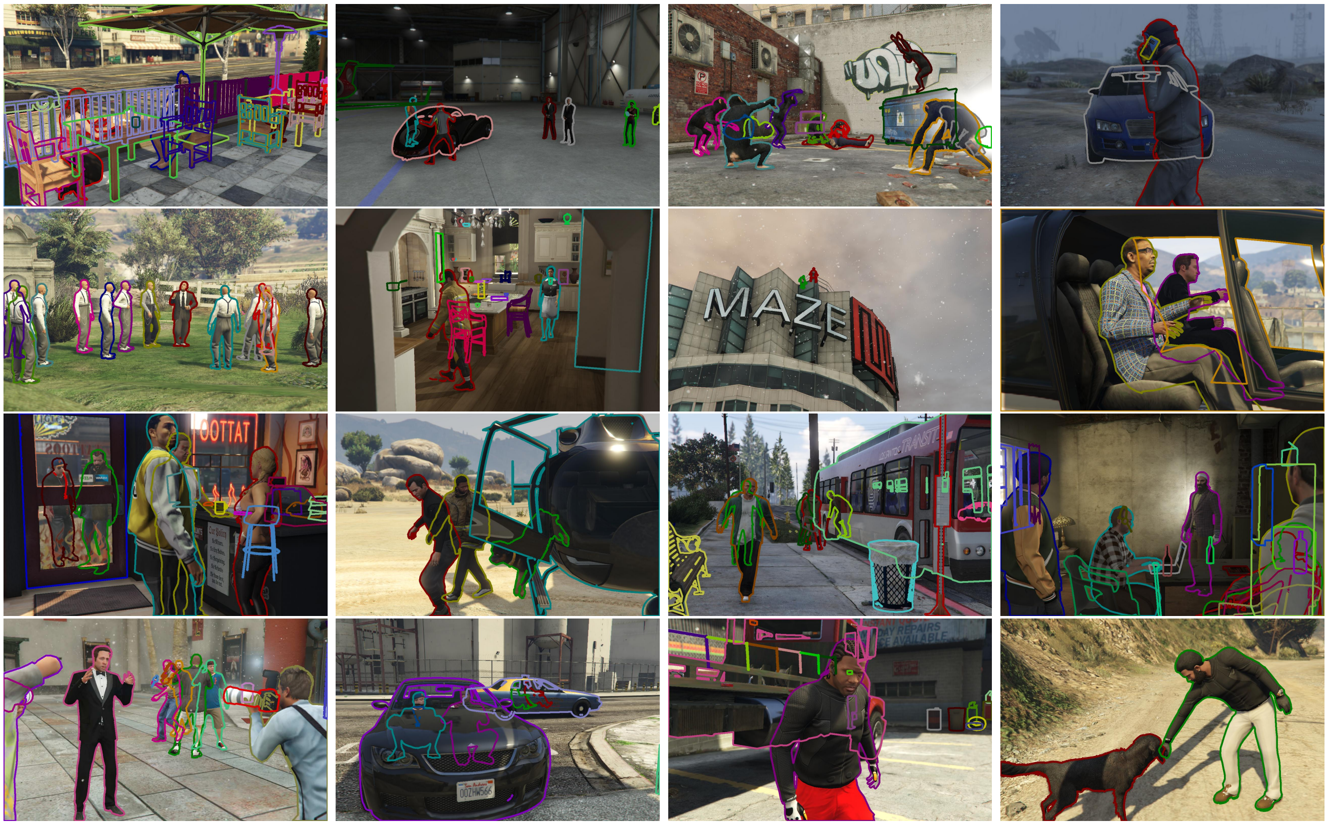

Figure 5: The proposed dataset contains diverse scenes, including outdoor scenes, indoor scenes, and different weather (sunny, rainy, storm). We also vary the appearance of the characters by changing their clothing as shown in the two images on the right in the bottom row.

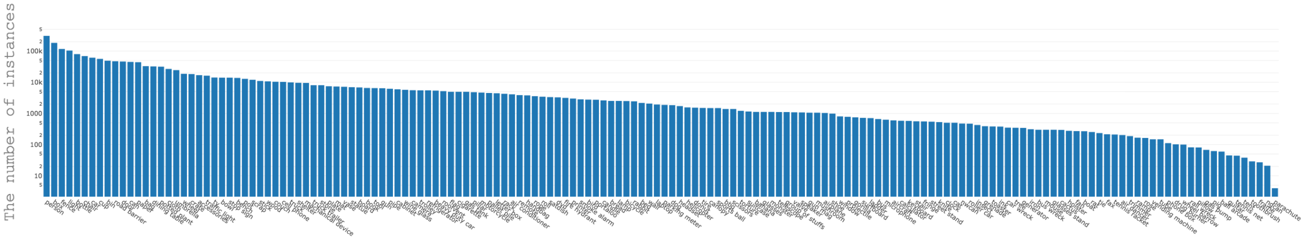

Figure 6: The number of instances per class in the proposed SAIL-VOS dataset. There are 162 semantic classes in total.

the label, we access the name of the 3D model file used to render the object. We use Script Hook V and [1] to get the memory address where the information related to the object is stored and extract the 3D model file name. The name usually contains the semantics of the object, *e.g.*, the model with the name `prop_laptop_01a` is a 3D model for a laptop. We use the names to group the models by comparing the similarity of their names. We further manually merge the groups with either similar semantics and/or shape into the final 162 classes. Of those 162 classes, 48 overlap with the MS-COCO dataset [65], *i.e.*, 60% of the classes in MS-COCO can be found in the proposed SAIL-VOS dataset. Classes that exist in MS-COCO but not in our dataset include `hair drier`, `giraffe`, and `kite`, and classes which exist in the SAIL-VOS dataset but not in MS-COCO include `lamp`, `pot` and `cabinet`.

Depth ordering: We use the depth buffer to compute the depth ordering by sorting the depth $d_f^o \; \forall o \in \mathcal{O}$ at each pixel. The depth ordering in the scene is computed at the pixel level based on the depth buffer. Note that the depth ordering is not at the object level.

Pose annotation: In addition to semantic instance level mask annotation, we also record the pose information. Specifically, we record the 2D and 3D pose information for people. This information includes the 2D coordinate on the image plane and the 3D position in the world coordinate for 30 facial landmarks, 18 points on the body and 30 points on both hands (15 points per hand). We use the Script Hook V library to retrieve the 2D and 3D pose information. For other objects including vehicles and properties, we record

Table 1: Amodal segmentation dataset statistics. Note, for computation of the number of occluded instance we define an object to be occluded if the occlusion rate is larger than 1%.

Dataset	COCOA	COCOA-cls	D2S	DYCE	Ours
Image/Video	Image	Image	Image	Image	Video
Resolution	275K pix	275K pix	3M pix	1M pix	1M pix
	-	-	1440×1920	1000×1000	800×1280
Synthetic/Real	Real	Real	Real	Synthetic	Synthetic
# of images	5,073	3499	5,600	5,500	111,654
# of classes	-	80	60	79	162
# of instances	46,314	10,562	28,720	85,975	1,896,295
# of occluded instances	28,106	5,175	16,337	70,766	1,653,980
Avg. occlusion rate	18.8%	10.7%	15.0%	27.7%	56.3%

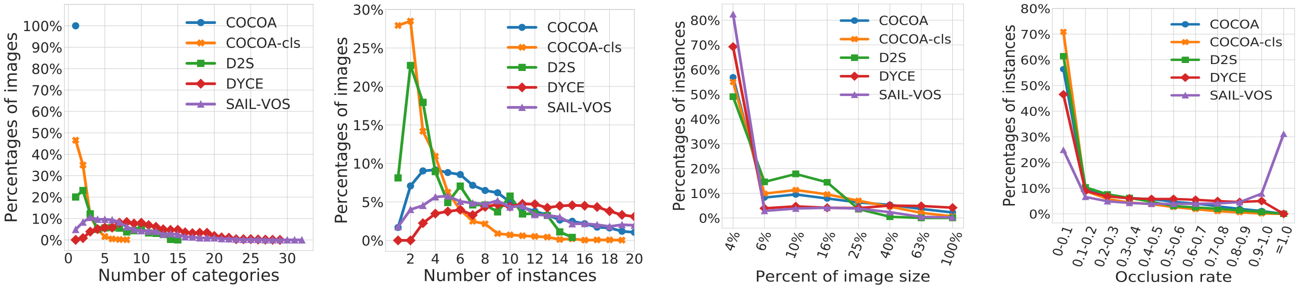

Figure 7: Comparison of the SAIL-VOS dataset with COCOA [110], COCOA-cls [110, 30], D2S [29, 30], and DYCE [24].

the 3D location in the world coordinate, the 3D rotation matrix, the 2D coordinate on the image plane and the 2D bounding box.

4. SAIL-VOS Dataset

We collect 111,654 frames in total from GTA-V at a resolution of 800 × 1280. A total of 201 video sequences are collected, covering different weather conditions, illumination, scenes and scenarios as illustrated in Fig. 5. The annotation includes semantic instance level segmentation and amodal segmentation. Moreover, each object is manually assigned a class label. There are a total of 162 classes defined in the proposed SAIL-VOS dataset, 48 of which overlap with the MS-COCO object classes. Moreover, each class is subdivided into subclasses to provide more fine-grained semantic information. For instance, the class `road barrier` is divided into 6 subclasses, containing `bollard`, `traffic cone`, `road pole`, *etc*. Every object is assigned a class label and a subclass label. We also have the 2D and 3D pose annotation exclusively for person.

In Fig. 6, we show the number of instances per class in the dataset. We found the dataset to have a similar distribution as the MS COCO dataset [65]. In Tab. 1 we compare the SAIL-VOS dataset with other amodal segmentation datasets, *i.e.*, COCOA [110], COCOA-cls [110, 30], DYCE [24] and D2S [29, 30], looking at the number of instances included in the dataset, the occlusion ratio and the resolution. In Fig. 7, we also show a detailed comparison among the amodal datasets, comparing the number of categories and instances per image, the size of the objects and the occlusion rate.

Dataset splits: We split the dataset into *training*, *validation* and test set based on the geographic location of the scenes. The *training set* contains 160 video sequences (84,781 images, 1,388,389 objects) while the *validation set* contains 41 (26,873 images, 507,906 objects). The portion of overlapping models between training and validation set is 32.9%, *i.e.*, there are only 32.9% of the models in the training set that also appear in the validation set. Note that the model defines the geometry but different textures may be used during rendering. In addition to the training and validation set, we retain a *test-dev* set and a *test-challenge* set for future use.

5. SAIL-VOS Problem Formulation

Because of annotations for modal and amodal semantic segmentation, human pose and depth ordering, a variety of tasks can be evaluated using the presented SAIL-VOS dataset. We discuss some of the possibilities next.

Due to the semantic segmentation labels, class-agnostic and class-specific modal and amodal instance level segmentation for video data can be assessed. Because of the temporal density, frame-based and tracking based formulations can be evaluated. Because the proposed dataset is syn-

Table 2: Segmentation performance on the SAIL-VOS dataset in the **class-agnostic** setting. Both the modal mask (visible mask) and amodal mask are evaluated. We report AP_{50} (average precision at IoU threshold 50%) and AP (average precision) using four methods.

	Modal mask							Amodal mask						
	AP_{50}	AP	AP^{P}_{50}	AP^{H}_{50}	AP^{L}_{50}	AP^{M}_{50}	AP^{S}_{50}	AP_{50}	AP	AP^{P}_{50}	AP^{H}_{50}	AP^{L}_{50}	AP^{M}_{50}	AP^{S}_{50}
MaskRCNN [38]	**40.6**	**28.0**	**51.2**	**13.5**	**74.6**	**20.2**	**5.6**	-	-	-	-	-	-	-
MaskAmodal [30]	-	-	-	-	-	-	-	40.4	**26.6**	**51.2**	14.8	72.9	**20.6**	6.8
MaskJoint	38.8	26.0	49.5	11.9	70.4	17.4	6.4	**40.8**	26.4	**51.2**	**15.8**	**73.1**	19.6	**7.5**

Table 3: Segmentation performance on the SAIL-VOS dataset in the **class-specific** setting. Both the modal mask (visible mask) and amodal mask are evaluated. We report AP_{50} (average precision at IoU threshold of 50%) and AP (average precision) using four methods.

	Modal mask							Amodal mask						
	AP_{50}	AP	AP^{P}_{50}	AP^{H}_{50}	AP^{L}_{50}	AP^{M}_{50}	AP^{S}_{50}	AP_{50}	AP	AP^{P}_{50}	AP^{H}_{50}	AP^{L}_{50}	AP^{M}_{50}	AP^{S}_{50}
MaskRCNN [38]	24.1	**14.3**	**24.7**	17.2	**42.8**	**21.3**	4.9	-	-	-	-	-	-	-
MaskAmodal [30]	-	-	-	-	-	-	-	23.0	13.0	**24.3**	16.7	36.6	**21.5**	**6.1**
MaskJoint	**24.5**	14.2	24.1	**17.6**	38.9	21.0	**5.1**	**24.8**	**14.1**	**24.3**	**18.9**	**37.8**	**21.5**	5.7

Table 4: IoU on the DAVIS validation set.

DAVIS fraction	0%	10%	20%	30%	50%	100%
VideoMatch-S	**0.74**	**0.77**	**0.78**	**0.78**	**0.78**	0.79
VideoMatch	0.55	0.66	0.73	0.74	**0.78**	**0.81**

thetic and due to available smaller real-world amodal image datasets, transferability can be measured.

Future possibilities of the collected dataset include reasoning about depth ordering, 2D and 3D pose estimation. We will not focus on those directions subsequently.

In the following we focus on class-agnostic and class-specific modal and amodal instance level video segmentation using frame-based techniques. We focus on frame-based techniques to ensure comparability with recently proposed existing amodal segmentation methods. We also evaluate transferability to assess whether training on synthetic data can improve video object segmentation. We defer an assessment of tracking based formulations to future work. Importantly, we make all the baselines, the dataset and hopefully an evaluation server available to the community.[1]

6. Experiments

In the following we first present evaluation metrics for class-agnostic and class-specific modal and amodal instance level segmentation using frame-based detection techniques, before discussing quantitative and qualitative results.

Evaluation Metrics: Since we focus on frame-based detection techniques we follow Pascal VOC [25] and MS-COCO [65] and use average precision (AP) as the evaluation metric for the modal segmentation and the amodal segmentation. The AP is computed by averaging the APs at increasing IoU thresholds from 0.5 to 0.95. We also report AP_{50}, which is the AP with an IoU threshold of 0.5.

To better assess a method we look at a variety of data splits. We report AP^{P}_{50} for objects with no or partial occlusion (occlusion rate less than 0.25) and AP^{H}_{50} for heavily occluded objects (occlusion rate larger than or equal to 0.25). Also, we report AP^{L}_{50}, AP^{M}_{50} and AP^{S}_{50} to evaluate the performance of segmenting large (area larger than 96^2), medium (area between 32^2 to 96^2) and small objects (area less than 32^2).

There are two common settings for evaluating amodal segmentation: the class-agnostic setting, *e.g.*, the COCOA dataset evaluation in [110] and the class-specific setting, *e.g.*, the COCOA-cls and D2S evaluation in [30]. In the class-agnostic setting, the network is trained to detect object segments without using the class label. In the class-specific setting, the network is trained to detect instance level semantic object segments. We evaluate the frame-based detection techniques in both of the settings on the proposed SAIL-VOS dataset. For the object-specific setting, for now, we focus on 24 classes in our dataset for simplicity.

Approaches: Irrespective of the class-agnostic or class-specific setting we evaluate three methods. First, we apply MaskRCNN [38] for predicting the modal masks, training on the SAIL-VOS training set and testing on the SAIL-VOS validation set. The baseline MaskAmodal is a MaskRCNN trained on the amodal masks of the SAIL-VOS dataset. This approach is tested and discussed in [30] and according to [30] MaskAmodal is the state-of-the-art method for predicting the amodal mask on the COCOA [110] and COCOA-cls [110, 30] datasets. Note that occluded objects can cause issues because the model simply assumes the occluder is the object of interest. MaskJoint aims to jointly predict the modal and amodal mask and is our extension of

[1] http://sailvos.web.illinois.edu

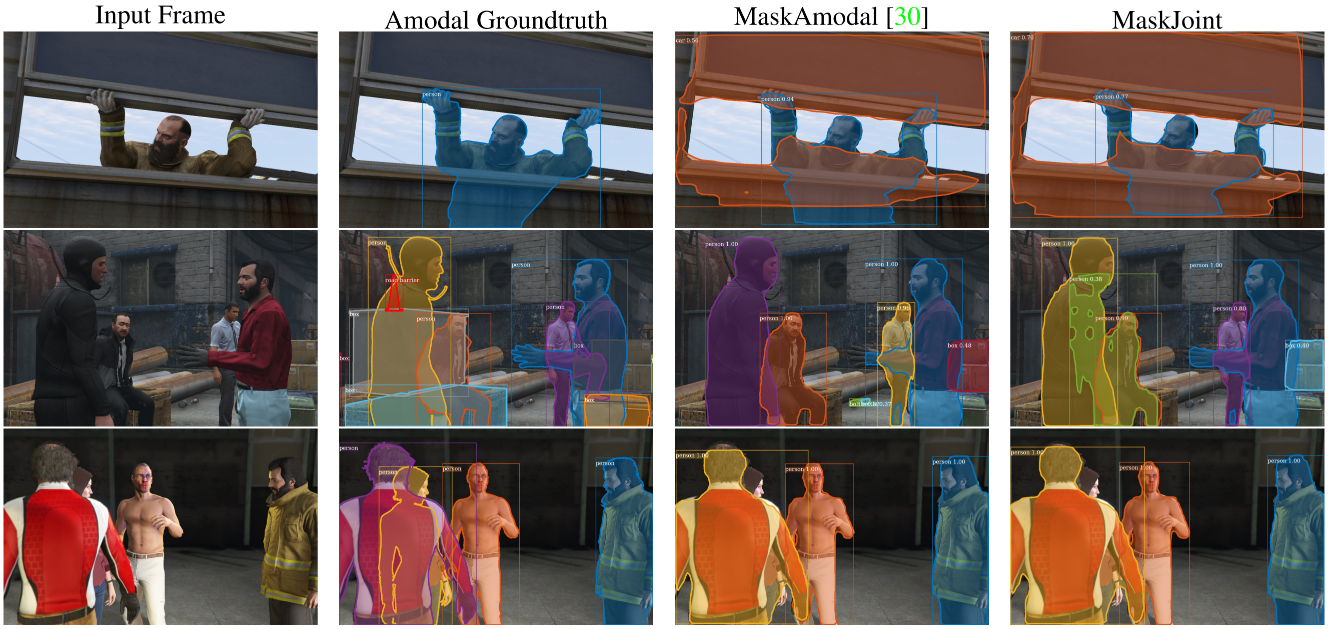

Figure 8: The qualitative results of the amodal mask of the baselines MaskAmodal [30], MaskCascade and MaskJoint on SAIL-VOS dataset.

the baseline MaskAmodal, adding an additional mask head to predict the modal mask. During training and testing, we exclude objects with occlusion rate larger than 75%.

Quantitative Results: We present results for the class-agnostic setting in Tab. 2. We observe that MaskRCNN performs best when predicting the modal segmentation in all cases. In contrast, MaskAmodal performs best when predicting the amodal segmentation on AP and medium objects while MaskJoint performs best on AP_{50}, objects with heavy occlusion, large and small objects.We present results for the class-specific setting in Tab. 3. We observe MaskRCNN to outperform other baselines in the modal case except for AP_{50}, objects with high occlusion and small objects, and MaskJoint outperforms MaskAmodal when predicting the amodal segmentation except for small objects.

Video object segmentation: To assess whether training with additional synthetic data provides improvements in accuracy, we evaluate on the DAVIS16 dataset [75]. We employ the video object segmentation baseline VideoMatch [46] in this experiment. We show results for training of VideoMatch with the proposed SAIL-VOS dataset (VideoMatch-S) and without using the discussed data (VideoMatch). Moreover we vary the percentage of the used DAVIS-2016 training data from 0% to 100% during training. We report the Intersection over Union (IoU) metric computed using the DAVIS16 validation set in Tab. 4. We found that the proposed synthetic dataset is useful when access to real data is limited. Specifically, without access to real data (0% DAVIS fraction), pretraining on the SAIL-VOS dataset improves the performance by 19% IoU, boosting the performance of VideoMatch from 55% to 74%.

Qualitative Results: We show qualitative results of the amodal segmentation for MaskAmodal and MaskJoint in Fig. 8. We observe that the baselines are able to reason about the object contour under slight occlusion. As shown in the first two rows of Fig. 8, the contour of the person and the box is inferred. However, predicting the amodal segmentation of objects under heavy occlusion remains challenging for the investigated approaches. All methods fail to segment the second person from the left in the third row of Fig. 8. In this example, as the visible region of the object is small, it is hard to segment the object using a frame-based technique, *i.e.*, predicting the amodal segmentation by only looking at one frame. This suggests that a video-based approach using the temporal context (*i.e.*, using adjacent frames for reasoning about the amodal mask) is a promising direction to improve amodal segmentation. The SAIL-VOS dataset can serve as a good test bed and training source for development of video-based techniques.

7. Conclusion

We propose a new synthetic dataset for semantic amodal instance level video object segmentation (SAIL-VOS) which has a compelling diversity and rich annotations. We hope to stimulate new research in a variety of directions and show that the dataset can be used to improve video segmentation if little real data is available.

Acknowledgments: This work is supported in part by NSF under Grant No. 1718221, Samsung, and 3M. We thank NVIDIA for providing GPUs used for this work.

References

[1] MapInfoTool. https://github.com/CamxxCore/MapInfoTool/. 5

[2] H. A. Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother. Augmented Reality Meets Deep Learning for Car Instance Segmentation in Urban Scenes. In *Proc. BMVC*, 2017. 2

[3] M. Andriluka, S. Roth, and B. Schiele. People-Tracking-by-Detection and People-Detection-by-Tracking. In *Proc. CVPR*, 2008. 2

[4] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour Detection and Hierarchical Image Segmentation. *PAMI*, 2010. 2

[5] A. Arnab and P. H. S. Torr. Pixelwise instance segmentation with a dynamically instantiated network. In *Proc. CVPR*, 2017. 2

[6] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation. In *Proc. CVPR*, 2017. 2

[7] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 2011. 3

[8] O. Barinova, V. Lempitsky, and P. Kohli. On detection of multiple object instances using hough transforms. *PAMI*, 2012. 2

[9] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *IJCV*, 1994. 3

[10] A. Blade. Script Hook V. http://www.dev-c.com/gtav/scripthookv/. 3

[11] S. Caelles, Y. Chen, J. Pont-Tuset, and L. Van Gool. Semantically-guided video object segmentation. *arXiv preprint arXiv:1704.01926*, 2017. 3

[12] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *Proc. CVPR*, 2017. 3

[13] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In *Proc. ICLR*, 2015. 2

[14] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. ECCV*, 2018. 2

[15] Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *Proc. CVPR*, 2018. 3

[16] Y.-T. Chen, X. Liu, and M.-H. Yang. Multi-instance object segmentation with occlusion handling. In *Proc. CVPR*, 2015. 2

[17] J. Cheng, Y.-H. Tsai, W.-C. Hung, S. Wang, and M.-H. Yang. Fast and accurate online video object segmentation via tracking parts. In *Proc. CVPR*, 2018. 3

[18] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. SegFlow: Joint learning for video object segmentation and optical flow. In *Proc. ICCV*, 2017. 3

[19] D. Comaniciu and P. Meer. Robust Analysis of Feature Spaces: Color Image Segmentation. In *Proc. CVPR*, 1997. 2

[20] M. Cordts, M. Omran, S. Ramos, R. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proc. CVPR*, 2016. 2

[21] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance-sensitive fully convolutional networks. In *Proc. ECCV*, 2016. 2

[22] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proc. CVPR*, 2016. 2

[23] S. Edelman and T. Poggio. Integrating Visual Cues for Object Segmentation and Recognition. *Optics News*, 1989. 2

[24] K. Ehsani, R. Mottaghi, and A. Farhadi. SeGAN: Segmenting and Generating the Invisible. In *Proc. CVPR*, 2018. 1, 2, 3, 6

[25] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010. 2, 7

[26] M. Fabbri, F. Lanzi, S. Calderara, A. Palazzi, R. Vezzani, and R. Cucchiara. Learning to Detect and Track Visible and Occluded Body Joints in a Virtual World. In *Proc. ECCV*, 2018. 3

[27] A. Faktor and M. Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014. 3

[28] A. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen. JumpCut: Non-Successive Mask Transfer and Interpolation for Video Cutout. In *Proc. SIGGRAPH*, 2015. 1, 3

[29] P. Follmann, T. Böttger, P. Härtinger, R. König, and M. Ulrich. MVTec D2S: Densely Segmented Supermarket Dataset. In *Proc. ECCV*, 2018. 1, 3, 6

[30] P. Follmann, R. König, P. Härtinger, and M. Klostermann. Learning to see the invisible: End-to-end trainable amodal instance segmentation. *arXiv preprint arXiv:1804.08864*, 2018. 1, 6, 7, 8

[31] G. Ghiasi, Y. Yang, D. Ramanan, and C. C. Fowlkes. Parsing Occluded People. In *Proc. CVPR*, 2014. 2

[32] A. Guisti, D. Ciresan, J. Masci, L. Gambardella, and J. Schmidhuber. Fast image scanning with deep max-pooling convolutional neural networks. In *Proc. ICIP*, 2013. 2

[33] R. Guo and D. Hoiem. Beyond the line of sight: labeling the under- lying surfaces. In *Proc. ECCV*, 2012. 1, 2

[34] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from RGB-D images. In *Proc. CVPR*, 2013. 1, 2

[35] A. Handa, T. Whelan, J. McDonald, and A. J. Davison. A Benchmark for RGB-D Visual Odometry, 3D Reconstruction and SLAM. In *Proc. ICRA*, 2014. 3

[36] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic Contours from Inverse Detectors. In *Proc. ICCV*, 2011. 2

[37] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *Proc. ECCV*, 2014. 2

[38] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *Proc. ICCV*, 2017. 2, 7

[39] X. He and S. Gould. An Exemplar-based CRF for Multi-instance Object Segmentation. In *Proc. CVPR*, 2014. 2

[40] X. He, R. S. Zemel, and M. A. Carreira-Perpiñán. Multiscale Conditional Random Fields for Image Labeling. In *Proc. CVPR*, 2004. 2

[41] D. J. Heeger. Model for the extraction of image flow. *JOSA*, 1987. 3

[42] B. K. P. Horn and B. G. Schunck. Determining Optical Flow. *AI*, 1981. 3

[43] E. Hsiao and M. Hebert. Occlusion Reasoning for Object Detection under Arbitrary Viewpoint. In *Proc. CVPR*, 2012. 2

[44] Y.-T. Hu, J.-B. Huang, and A. Schwing. Unsupervised video object segmentation using motion saliency-guided spatio-temporal propagation. In *Proc. ECCV*, 2018. 3

[45] Y.-T. Hu, J.-B. Huang, and A. G. Schwing. MaskRNN: Instance Level Video Object Segmentation. In *Proc. NIPS*, 2017. 3

[46] Y.-T. Hu, J.-B. Huang, and A. G. Schwing. VideoMatch: Matching based video object segmentation. In *Proc. ECCV*, 2018. 3, 8

[47] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang. DeepMVS: Learning multi-view stereopsis. In *Proc. CVPR*, 2018. 3

[48] S. D. Jain, B. Xiong, and K. Grauman. FusionSeg: Learning to combine motion and appearance for fully automatic segmention of generic objects in videos. *Proc. CVPR*, 2017. 3

[49] V. Jampani, R. Gadde, and P. V. Gehler. Video propagation networks. In *Proc. CVPR*, 2017. 3

[50] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In *Proc. ICRA*, 2017. 3

[51] B. Kaneva, A. Torralba, and W. T. Freeman. Evaluation of image features using a photorealistic virtual world. In *Proc. ICCV*, 2011. 3

[52] A. Kar, S. Tulsiani, J. Carreira, and J. Malik. Amodal completion and size constancy in natural scenes. In *Proc. ICCV*, 2015. 1, 2, 3

[53] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele. Lucid data dreaming for object tracking. *arXiv preprint arXiv:1703.09554*, 2017. 3

[54] A. Khoreva, F. Perazzi, R. Benenson, B. Schiele, and A.Sorkine-Hornung. Learning video object segmentation from static images. In *Proc. CVPR*, 2017. 3

[55] G. Kim and E. P. Xing. On multiple foreground cosegmentation. In *Proc. CVPR*, 2012. 2

[56] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother. Instancecut: from edges to instances with multicut. In *Proc. CVPR*, 2017. 2

[57] S. Konishi and A. L. Yuille. Statistical cues for domain specific image segmentation with performance analysis. In *Proc. CVPR*, 2000. 2

[58] P. Krḧenbühl. Free supervision from video games. In *Proc. CVPR*, 2018. 3

[59] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic Models for segmenting and labeling sequence data. In *Proc. ICML*, 2001. 2

[60] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV Workshop*, 2004. 2

[61] F. Li, T. Kim, A. Humayun, D. Tsai, and J. Rehg. Video segmentation by tracking many figure- ground segments. In *Proc. ICCV*, 2013. 1, 3

[62] K. Li and J. Malik. Amodal Instance Segmentation. In *Proc. ECCV*, 2016. 1, 2

[63] X. Li, Y. Qi, Z. Wang, K. Chen, Z. Liu, J. Shi, P. Luo, C. C. Loy, and X. Tang. Video object segmentation with re-identification. *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*, 2017. 3

[64] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In *Proc. CVPR*, 2017. 2

[65] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. In *Proc. ECCV*, 2014. 2, 5, 6, 7

[66] S. Liu, J. Jia, S. Fidler, and R. Urtasun. SGN: Sequential grouping networks for instance segmentation. In *Proc. ICCV*, 2017. 2

[67] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Proc. CVPR*, 2015. 2

[68] M. Maire, S. X. Yu, and P. Perona. Hierarchical scene annotation. In *Proc. BMVC*, 2013. 1, 2

[69] N. Mayer, E. Ilg, P. Fisher, C. Hazirbas, D. Cremers, A. Dosovitskiy, and T. Brox. What Makes Good Synthetic Training Data for Learning Disparity and Optical Flow Estimation? *https://arxiv.org/abs/1801.06397*, 2018. 3

[70] G. Neuhold, T. Ollmann, S. R. Rota, and P. Kontschieder. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In *Proc. ICCV*, 2017. 2

[71] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *PAMI*, 2014. 1, 3

[72] Y.-I. Ohta, T. Kanade, and T. Sakai. An Analysis System for Scenes Containing Objects with Substructures. In *IJCPR*, 1978. 2

[73] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *Proc. ICCV*, 2013. 3

[74] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Occlusion Patterns for Object Class Detection. In *Proc. CVPR*, 2013. 2

[75] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proc. CVPR*, 2016. 8

[76] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016. 1, 3

[77] P. O. Pinheiro, R. Collobert, and P. Dollár. Learning to segment object candidates. In *Proc. NIPS*, 2015. 2

[78] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *Proc. ECCV*, 2016. 2

[79] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 1, 3

[80] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *Proc. CVPR*, 2012. 1, 3

[81] S. R. Richter, Z. Hayder, and V. Koltun. Playing for benchmarks. In *Proc. ICCV*, 2017. 3

[82] S. R. Richter and S. Roth. Discriminative shape from shading in uncalibrated illumination. In *Proc. CVPR*, 2015. 3

[83] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for Data: Ground Truth from Computer Games. In *Proc. ECCV*, 2016. 3

[84] H. Riemenschneider, S. Sternig, M. Donoser, P. M. Roth, and H. Bischof. Hough regions for joining instance localization and segmentation. In *Proc. ECCV*, 2012. 2

[85] G. Seguin, P. Bojanowski, R. Lajugie, and I. Laptev. Instance-level video segmentation from object tracks. In *Proc. CVPR*, 2016. 3

[86] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In *Proc. ICLR*, 2014. 2

[87] T. Sharp, C. Keskin, D. P. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, D. Freedman, P. Kohli, E. Krupka, A. W. Fitzgibbon, and S. Izardi. Accurate, robust, and flexible real-time hand tracking. In *Proc. CHI*, 2015. 3

[88] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *PAMI*, 2000. 2

[89] J. Shotton, R. B. Girshick, A. W. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Crminisi, A. Kipman, and A. Blake. Efficient human pose estimation from single depth images. In *2013*, PAMI. 3

[90] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation. In *Proc. ECCV*, 2006. 2

[91] N. Silberman, D. Hoiem, , and R. Fergus. Indoor segmentation and support inference from RGBD images. In *Proc. ECCV*, 2012. 2

[92] N. Silberman, L. Shapira, R. Gal, and P. Kohli. A contour completion model for augmenting surface reconstructions. In *Proc. ECCV*, 2014. 1, 2

[93] N. Silberman, D. Sontag, and R. Fergus. Instance Segmentation of Indoor Scenes using a Coverage Loss. In *Proc. ECCV*, 2014. 2

[94] J. Tighe, M. Niethammer, and S. Lazebnik. Scene Parsing with Object Instances and Occlusion Ordering. In *Proc. CVPR*, 2014. 2

[95] P. Tokmakov, K. Alahari, and C. Schmid. Learning motion patterns in videos. In *Proc. CVPR*, 2017. 3

[96] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. In *Proc. ICCV*, 2017. 3

[97] D. Tsai, M. Flagg, and J. Rehg. Motion coherent tracking with multi-label mrf optimization. In *Proc. BMVC*, 2010. 1, 3

[98] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. *BMVC*, 2017. 3

[99] B. Wu and R. Nevatia. Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. *IJCV*, 2009. 2

[100] S. Wug Oh, J.-Y. Lee, K. Sunkavalli, and S. Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *Proc. CVPR*, 2018. 3

[101] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang. Youtube-vos: A large-scale video object segmentation benchmark. In *Proc. ECCV*, 2018. 1, 3

[102] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. Efficient video object segmentation via network modulation. In *Proc. CVPR*, 2018. 3

[103] Y. Yang, S. Hallman, D. Ramanan, and C. C. Fowlkes. Layered object models for image segmentation. *PAMI*, 2012. 2

[104] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *Proc. CVPR*, 2012. 2

[105] J. S. Yoon, F. Rameau, J. Kim, S. Lee, S. Shin, and I. S. Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *Proc. ICCV*, 2017. 3

[106] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell. BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling. In *https://arxiv.org/abs/1805.04687*, 2018. 2

[107] H. Zhang, T. Fang, X. Chen, Q. Zhao, and L. Quan. Partial similarity based nonparametric scene parsing in certain environment. In *Proc. CVPR*, 2011. 2

[108] Z. Zhang*, A. G. Schwing*, S. Fidler, and R. Urtasun. Monocular Object Instance Segmentation and Depth Ordering with CNNs. In *Proc. ICCV*, 2015. * equal contribution. 2

[109] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional Random Fields as Recurrent Neural Networks. In *Proc. ICCV*, 2015. 2

[110] Y. Zhu, Y. Tian, D. Metaxas, and P. Dollár. Semantic amodal segmentation. In *Proc. CVPR*, 2017. 1, 2, 6, 7