AMODAL VIDEO INSTANCE SEGMENTATION

BY

MINGXI SUN

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Adviser:

Alexander Schwing

ABSTRACT

TODO: Thanks

ACKNOWLEDGMENTS

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Human perception naturally reasons about information even if it is not fully visible [**?**]. The ability to reason about occlusions permits reaction based on the invisible. For instance, when switching lanes on a highway, a partial view of a car and its movement through the rear-window is enough to understand the situation by completing the occluded parts subconsciously. This in turn helps to prevent collisions.

Motivated by this capability, prior works [**?** **?** **?** **?** **?** **?** **?**] studied the problem of amodal segmentation from a single image, *i.e.*, the task of segmenting both the visible *and occluded* parts of an object. In meticulous work, Hu *et al.* [**?**] collected a dataset for the task of Semantic Amodal Instance Level Video Object Segmentation (SAIL-VOS). Annotation of video data permits to expand amodal reasoning to the time dimension, which also seems crucial for human perception. However, the models proposed by Hu *et al.* [**?**] do not take advantage of the temporal information and are mostly adapted from the modal instance segmentation task. While this is a very valuable first step, it is suboptimal for three reasons illustrated in. Specifically, missing temporal information prevents use of motion cues. Moreover, bounding boxes of amodal segmentations overlap much more significantly than those of modal masks. We found special treatment of this issue to improve results. In addition, amodal segmentation requires to deal with occlusions, *i.e.*, object information from observations needs to be propagated more broadly.

To tackle these three challenges, Yeh and I proposed Amodal-Net, a framework for SAIL-VOS with a flow based temporal backbone, a redesigned box-head and a revised mask-head. More specifically, our temporal backbone enables the model to reason about occluded regions based on current and past frames. For this we incorporate temporal information into the amodal prediction model. Our box-head utilizes a cascade architecture with soft Non-Maximum Suppression (NMS) to address the challenge of heavily overlapping amodal boxes. Lastly, we developed a mask head with a large receptive field and self-attention to better propagate object information far into the occluded regions.

Recently, Hu *et al.* [**?**] also collected detailed 3D information on SAIL-VOS. Intuitively, 3D information combined with temporal information will offer more clue for objects and their occlusion. Therefore, this thesis explores the effect of adding 3D information to the video data. In particular, the model will incorporate depth map and camera intrinsic and extrinsic matrices into the input, and perform 3D reprojection in the feature space.

[TODO: modify the result section and add new result] We evaluate our approach on the SAIL-VOS dataset [**?**], where we outperform state-of-the-art by 3.5% (absolute-gain)

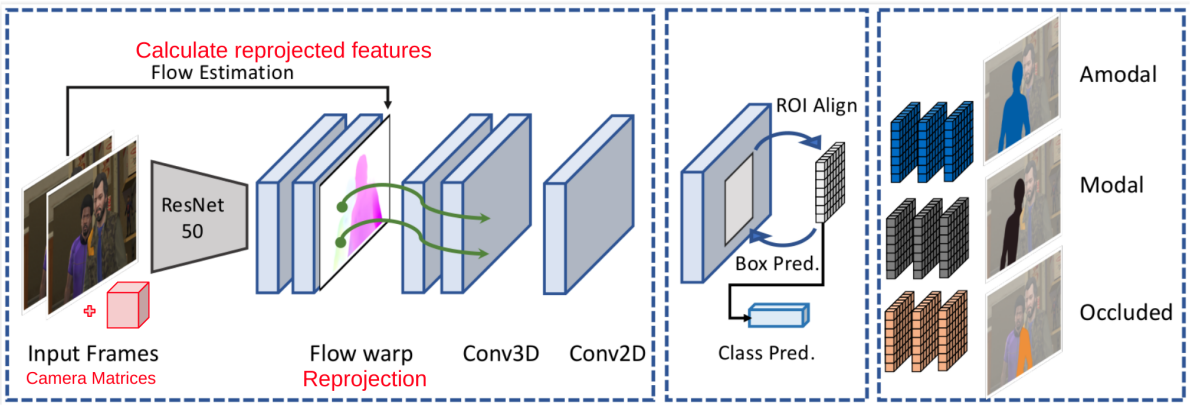

Figure 1.1: Overview of our proposed framework for learning SAIL-VOS. Our Amodal-Net includes a temporal backbone, an iterative box-head and mask-head, while it is trained jointly with the task of modal and occlusion mask prediction.

in Average Precision (AP) on amodal video instance segmentation. Next, to assess the generality of *non-temporal* components in the base model, we also report results on the COCO-Amodal dataset [**?**] and the KINS dataset [**?**]. Our approach outperforms state-of-the-art by 4.0% AP and 1.1% AP respectively.

CHAPTER 2: BACKGROUND

In this chapter I will breifly review relevant concepts. First I will review works on instance segmentation, which are often a source of inspiration for amodal methods. Subsequently I will discuss amodal segmentation. Lastly, I will review network architectures for extracting features from videos in other tasks, which serves as an inspiration for temporal data.

2.1 INSTANCE SEGMENTATION

Instance segmentation is a long-standing goal in computer vision that requires the prediction of object instances and their per-pixel segmentation mask. This makes it a hybrid of semantic segmentation and object detection. [**?** **?** **?** **?** **?** **?** **?** **?** **?** **?** **?** **?** **?** **?**].[TODO: maybe add more bg in instance segm and a picture that compares it with semestic segm]

Mask-RCNN-based methods have shown strong results on instance segmentation [**?** **?** **?** **?**]. These detect-then-segment methods first extract image features via a backbone network, *e.g.*, a ResNet. A Region Proposal Network (RPN) [**?**] subsequently retrieves a set of candidate bounding boxes and their objectness scores. This is achieved by sliding a small convolutional network over the features extracted via the backbone network.

These candidates may heavily overlap with each other. To reduce redundant computation due to the overlap, Non-Maximum Suppression (NMS) or its soft-version Soft-NMS [**?**] is applied to filter candidates.

Given the candidate bounding boxes, ROIAlign [**?**] is used to extract a feature map for each of the candidate boxes. The bounding box features are subsequently processed by a box-head and/or a mask-head, which regress to bounding box and segmentation mask respectively. The box-head consists of fully connected layers and yields a classification prediction and a corresponding bounding box. The mask-head consists of a stack of convolutional layers, which yield a 28×28 class-specific mask.

Variants of Mask-RCNN rely on a multi-stage refinement approach, *i.e.*, a cascade is used to enhance the box-prediction accuracy [**?** **?**]. While our method also relies on a cascade design, our model is specifically designed for the task of SAIL-VOS. Different from these works, we propose a temporal backbone to aggregate information over time, Soft-NMS to handle overlapping amodal boxes, and an iterative mask-head with attention to propagate information into occluded regions.

2.2 AMODAL IMAGE/INSTANCE SEGMENTATION.

Amodal instance segmentation is the task to delineate objects and their occluded parts in video or image data. The goal is to predict a mask for every object that is in the image, and the mask would draw out the whole object includding any part that may be occluded.

Annotations for amodal segmentation are challenging to obtain due to ambiguities caused by occlusions. Early works [**?** **?** **?** **?**] view the problem as a contour completion task. Other works [**?** **?** **?** **?** **?**] formulate the problem as occlusion reasoning from a single image. Generally, the focus is on the class-agnostic setting, where the predicted segmentation consists of only two classes, either foreground or background. Maire *et al*. [**?**] collect one of the earliest datasets for amodal segmentation, labeling 100 images.

More recently, datasets with class annotations were released. For instance, Zhu *et al*. [**?**] introduce the COCO-A dataset, which adds amodal annotations to 5000 images from the COCO dataset [**?**]. Even more recently, larger datasets for amodal segmentation have been collected. For example, Qi *et al*. [**?**] collect amodal annotations for the KITTI dataset [**?**]. Concurrently, Hu *et al*. [**?**] propose to use a game engine to automatically collect realistic annotations for amodal segmentation of video data. The use of a game engine circumvents any accuracy concerns for amodal annotation, which is challenging to obtain due to ambiguity. It hence avoids labor-intensive human labeling. Work by Hu *et al*. [**?**] further enables to train and evaluate amodal instance segmentation on videos, *i.e*., the task of SAIL-VOS. Based on these datasets, Mask-RCNN-based methods for single image instance segmentation have been proposed [**?** **?**]. More specifically, Follman *et al*. [**?**] discuss a two mask-head approach, which yields the amodal mask, the modal mask, and a combination of both results to form the occlusion mask for end-to-end training on all three masks. Hu *et al*. [**?**] propose to jointly train the amodal and modal mask. Different from these works, we consider input of videos and specifically design an architecture to address prevalent challenges for the task of SAIL-VOS.

2.3 VIDEO AND REPROJECTION ARCHITECTURES

Capturing temporal information is also crucial for tasks involving any form of video understanding, *e.g*., video segmentation, recognition, inpainting, *etc*. There are various methods to align features from different timestamps.For example, optical flow has been used as an additional deep-net input for video action recognition [**?** **?**]. Other methods incorporate optical flow by warping images or features, aligning them to the current frame of interest [**?** **?** **?**].In my previous work with Yeh, similar flow-based method was incorporated into the

net. Differently from those methods, in this thesis we will use reprojection in the net for alignment. [TODO: have a few more sentences on reprojection]

CHAPTER 3: APPROACH

The first step of our approach is Amodal-net, an amodal framework illustrated in **??** without the red annotations. The next step is to try to incorporate 3D information into the data used for training, as shown in red annotations in **??**. Intuitively, incorporating 3D information should allow the learning pipeline to use the temporal information better. For example, if the pipeline can see that an object is partialy blocked by an object that has a z value, it should know to still map out the object in its whole shape in the amodal mask.

The SAILVOS dataset[**?**] have annotations for depth map. SAIL-VOS 3D also from Hu *et al.* [**?**] also has camera intrinsics and extrinsics stored in the object files. Although there are some scene mismatches in the unit of a few mili-seconds, I will use these annotations to do mask reprojections in the data training pipeline.

3.1 OVERVIEW

Given a sequence of T images $\boldsymbol{I}_{t-T:t} = (\boldsymbol{I}_{t-T+1}, \ldots, \boldsymbol{I}_t)$ our goal is to predict for each object $o \in \mathcal{O}_t$ in the current frame $\boldsymbol{I}_t$ the corresponding amodal mask $\boldsymbol{M}_{t,o}$. We let $\mathcal{O}_t$ denote the set of detected objects in frame $\boldsymbol{I}_t$, while $\mathcal{M}_t = \{\boldsymbol{M}_{t,o}\ \forall o \in \mathcal{O}_t\}$ refers to the set of segmentation masks.

To accomplish this goal, we first extract features ϕ_t for all T frames in $\boldsymbol{I}_{t-T:t}$. Next, reprojection is used to spatially align features with the current frame $\boldsymbol{I}_t$ via warping. We then perform spatial and temporal aggregation to compute the feature Φ_t. This ensures that the backbone feature Φ_t summarizes temporal information.

Next, our cascade Soft-NMS box-head detects objects and crops Φ_t to extract object-level features $\Phi_{t,o}$ for each detected object $o \in \mathcal{O}_t$. Our box-head uses soft-thresholding during non-maximum suppression to better handle overlapping boxes. Given the object-level features $\Phi_{t,o}$, the amodal mask $\boldsymbol{M}_{t,o}$ for each object is predicted using an iterative mask-head. We incorporate a large receptive field and self-attention into the iterative mask-head. Because of this, information can propagate across the entire detection during mask prediction.

3.2 AMODAL-NET

The base model is Amodal-Net builds on top of Mask R-CNN[**?**]. Given the candidate bounding boxes, ROIAlign [**?**] is used to extract a feature map for each of the candidate

boxes. The bounding box features are subsequently processed by a box-head and/or a mask-head, which regress to bounding box and segmentation mask respectively. The box-head consists of fully connected layers and yields a classification prediction and a corresponding bounding box. The mask-head consists of a stack of convolutional layers.

Variants of Mask-RCNN rely on a multi-stage refinement approach, *i.e.*, a cascade is used to enhance the box-prediction accuracy [**?** **?**]. While our method also relies on a cascade design, our model is specifically designed for the task of SAIL-VOS. Amodal-net that Yeh and I developed with collaborator uses a temporal backbone to aggregate information over time, Soft-NMS to handle overlapping amodal boxes, and an iterative mask-head with attention to propagate information into occluded regions, as shown in **??**. We also found that multi-task training with the occlusion prediction task further improves performance. Combining the aforementioned techniques results in the following framework, for which a pictorial sketch is shown in Fig. **??**. This multi-task training is only included in the experiments for the Amodal-net and not included in the experiments with 3D Reprojection due to limitations in computation resources.

The changes in Amodal-net is made to address three challenges in this task: **(i)** a limited use of temporal information; **(ii)** a missing mechanism to handle heavily overlapping amodal boxes; and **(iii)** a propagation of object observations which is often too short-sighted. Amodal-net is based on **(i)** a temporal backbone which aggregates information across video frames, **(ii)** a box-head which better adjusts to overlapping detection boxes by using a cascade architecture with Soft-NMS, and **(iii)** a mask-head with increased receptive field and self-attention to propagate observations more broadly. Each of these components addresses the corresponding challenge.

3.3 3D REPROJECTION

Following the aforementioned intuition, 3D reprojection is added in the pipeline to align inputs in the temporal dimention, as shown in red in **??**. In the input, along with the images from t and $t-1$ frames, the camera matrices and the depth map of the corresponding frames is also passed in. The camera matrices include the intrinsic and extrinsic matrices of the camera.

The depth map and the camera matrices together is used to do reprojections from one frame to another. Standard computer vision reprojection algorithm warps an image from the perspective of one image to the perspetive of another image.

Given the image and the depth value of every pixel, we can get the camera coordinates for each pixel X_{cam}. For example, **??** plots one image in camera coordinates in 3D. Then it

is possible to calculate the camera coordinates of every pixel in another frame by using the camera matrices.

$$X_{cam} = K[R|t]X_{world}$$

where K is the intrinsic matrix and $[R|t]$ is the extrinsic matrix representing rotation and translation of the camera. If we have two cameras looking at the same scene,

$$X_{cam2} = K_2[R_2|t_2](K_1[R_1|t_1])^{-1}X_{cam1}$$

Finally, we can use X_{cam2} to recover the pixel coordinates and thus get the reprojected image in 2 dimensions. One complication in implementation is that initially I wasn't sure what format the depth map is stored in. After consulting with authors of [**?**] and some exploration, I discovered that the data is stored in a variation of NDC format. So camera coordinates of the image X_{cam} had to be computed from NDC coordinates X_ndc. The details of the implementation is **??**.

I also experimented with estimating the camera matrices from 2D and 3D points correspondence. However, due to the nature of how the data is collected in GTA-V and the format of the depth map. The estimated camera matrices induces a large error in reprojection compared to groundtruth camera matrices. In the end, I decided to use the groundtruth matrices recorded.

We can perform reprojeciton in image frames or masks. **??** shows an example of the reprojected image frame in full, and **??** shows some examples of reprojected masks. In these plots, the first picture is the groundtruth image of frame t; the second picture is the groundtruth image of frame $t-2$; the third picture is the groundtruth mask of an object at frame t; the fourth picture is the mask of the object at $t-2$ reproject into the perspective of frame t; the last picture of the mask of the object at frame $t-2$ directly. The caption shows intersection over union of the last two masks with the groundtruth mask *i.e.* the third mask. The reprojected image and mask should provide accurate information of the scene in previous frames since reprojection compensates the movement of the camera. But there is one drawback of this approach: it does not take into account the movement of objects in the scene. If the object is also moving, the reprojected image would not reflect that. But we can see from the captions of **??**, in general reprojected mask is closer to the groundtruth mask than directly using the mask from another frame, especially for objects that are still.

In the network, we use the same algorithm but perform reprojection on the features of images produced by the backbone. By doing so, the features from different temporal indices are aligned in the sense that there are from the same perspective, and each pixel corresponds

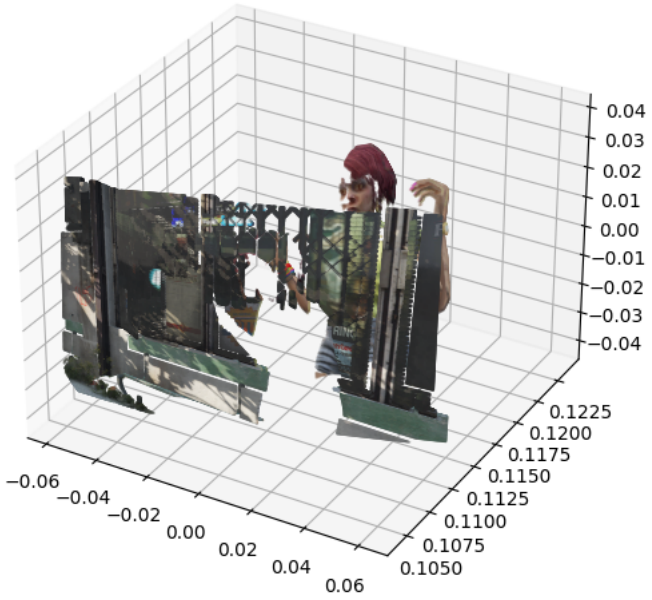

Figure 3.1: Projection of a frame in 3D camera coordinates

to the same spatial location. For this purpose, the depth map is reshaped into different shapes in the network to match the dimension of the features in the feature pyramid.

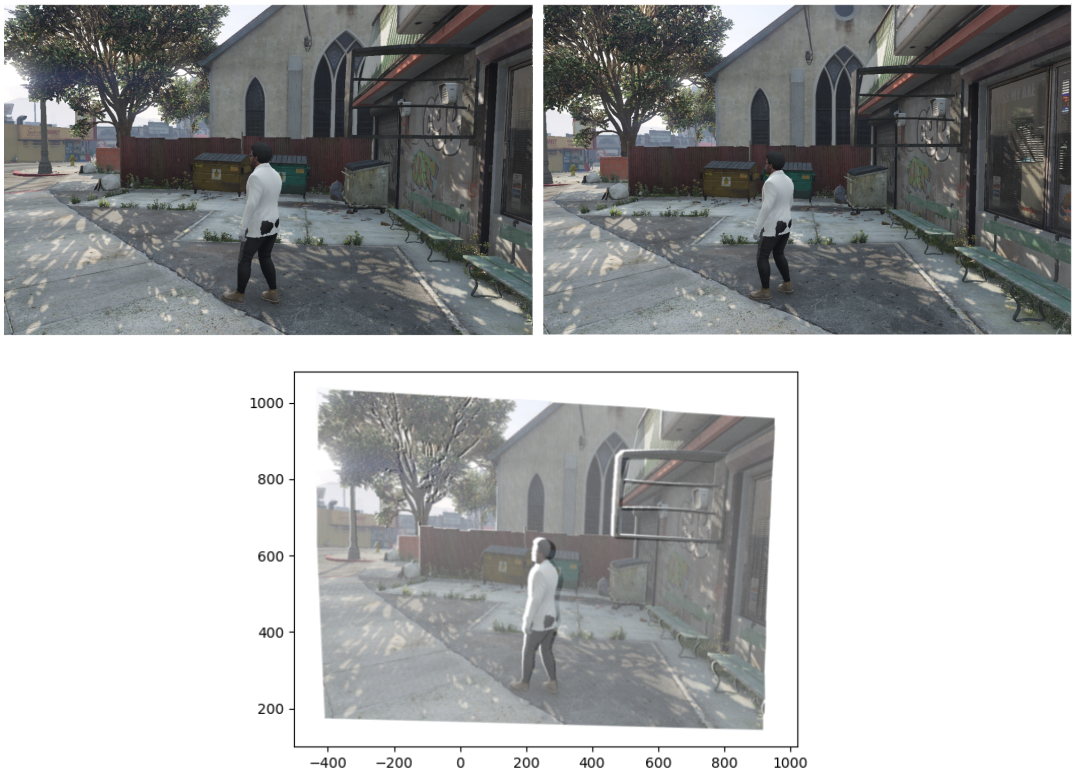

Figure 3.2: Image from SAILVOS dataset video tonya_concat_1 frame 270 and 271. The third figure is the result of reproject frame 270 to the view of 271. We can see that the camera moved left from frame 270 to 271, which is reflected in the reprojected image.

CHAPTER 4: EXPERIMENTS

4.1 METRICS AND BASELINE

To evaluate amodal segmentation, we report the commonly used average precision (AP) averaged over IoU thresholds from 50% to 95%. We also report AP with an IoU threshold of 50%, *i.e.*, AP_{50}. To further study the results, we compute a range of metrics, including AP^{P}_{50} and AP^{H}_{50}. Both report the AP_{50} using a subset of instances containing (P)artial ($< 25\%$) or (H)eavy ($\geq 25\%$) occlusions. Similarly, we also report across different instance sizes, AP^{L}_{50}, AP^{M}_{50}, and AP^{S}_{50} which correspond to pixel area of (L)arge ($\geq 96^2$), (M)edium ($[32^2, 96^2]$), and (S)mall ($\leq 32^2$) box areas respectively.

We compare our approach to two recent Mask-RCNN-based amodal segmentation methods, *MaskAmodal* [**?**] and *MaskJoint* [**?**]. MaskAmodal directly trains the Mask-RCNN on the task of amodal mask prediction. Differently, MaskJoint learns both amodal and model mask prediction simultaneously by introducing another mask-head into Mask-RCNN.

4.2 SAIL-VOS DATASET

The SAIL-VOS dataset consists of 160 training and 41 validation video sequences with $800 \times 1,280$ resolution images annotated with amodal/modal boxes and segmentation masks. Following Hu *et al.* [**?**], objects with occlusion rate larger than 75% are excluded from training and testing. We consider two common experimental settings: the class-specific setting which focuses on a 24 class subset within the dataset, and a class-agnostic setting which disregards the class-labels and views all objects to be of a single class.

4.3 AMODAL-NET EXPERIMENTS

We report quantitative results in Tab. **??**. All the results are reported using $T = 2$ frames. We also experimented with larger history. However, results did not change compared to using two frames. We suspect that usefulness of optical flow degrades as the history increases.

As shown in Tab. **??**,Amodal-net outperforms baselines [**?**] by 3.5% AP in the class-specific setting and by 3.6% AP in the class-agnostic setting. We also observe gains on the other metrics except for AP^{H}_{50} in the class-agnostic setting. These results validate that the proposed backbone and the box/mask-head tailored for amodal segmentation are effective and improve results.

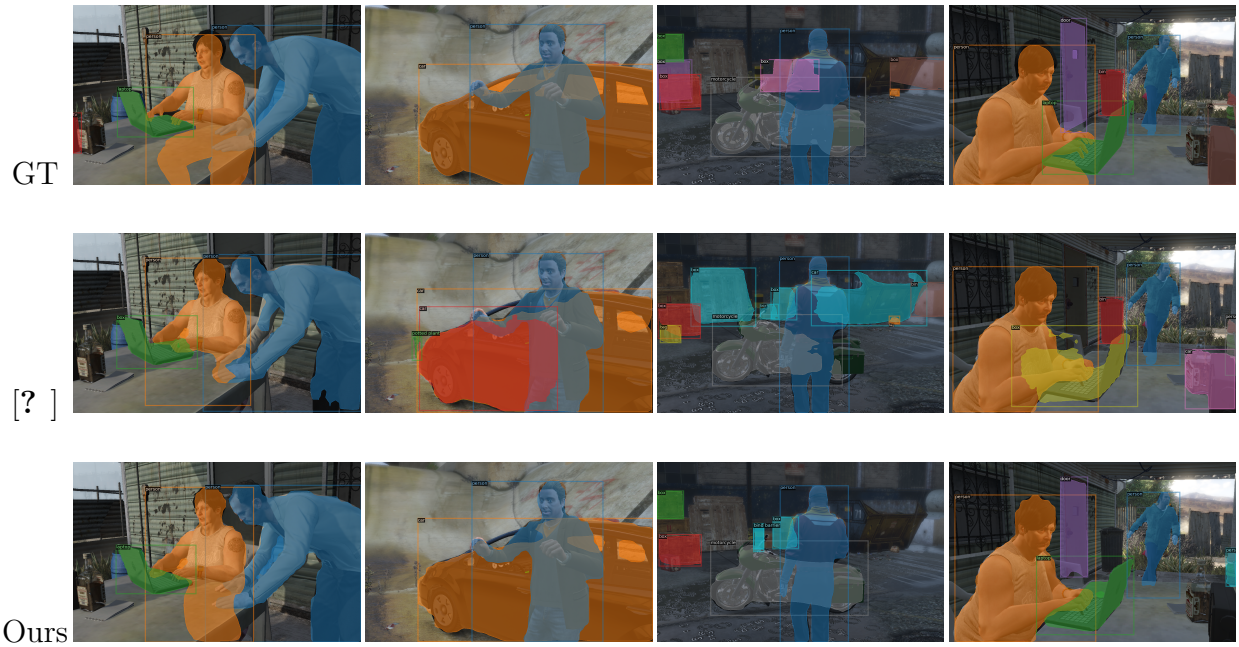

Figure 4.1: Qualitative comparison with [**?**] on SAIL-VOS dataset in the class-specific setting.

We provide qualitative results in Fig. **??**. Note that our approach successfully predicts the amodal mask despite occlusions. In column 1, half of the person is occluded by a table. The model correctly infers the lower half of the person. In column 2, our approach correctly predicts the overlapping amodal boxes, inferring a car and a person. In column 3, we successfully segment the entire motorcycle, propagating information 'through' the person. In column 4, the segmentations of the laptop and person correctly maintain their corresponding boundaries.

Next, we conduct an ablation study to assess the merits of the proposed components. Tab. **??** shows that each of the proposed components leads to improvements in the amodal mask's AP. In row 2, we validated that multi-task training with occlusion annotations (Sec. **??**) is beneficial. To experiment with different numbers of mask layers, we freeze the box-branch and only train the amodal mask. In row 3 and 4, we observe that using nine mask layers achieves the best results and adding more layers doesn't improve further. In row 5, we validated that the use of flow (Sec. **??**) is effective. In row 6 and 7, we see that the cascade box regression along with Soft-NMS (Sec. **??**) leads to improvements in box AP. Lastly, in row 8, further refinement with mask iterations (Sec. **??**) also improves the amodal segmentation's accuracy.

4.4 SANITY CHECK EXPERIMENTS

Before add reprojection to the network, I conducted some sanity check experiments to evaluate the effect of reprojection and catch any mistake there is.

My first experiment is to evaluate the gain on AP from doing reprojection. Here are the quantative results in table 1. The first three lines are from my previous project on this. The last four lines are the evaluation of using groundtruth masks from $t-1$ or $t-2$ as prediction, with or without reprojection. As one would expect, the accuracy of the lines that is using reprojection is higher, confirming our pressumption that reprojection should allows us to use 3D and temporal information better. But obviously these lines are using modified version of groundtruth masks, so it is not comparable to the first three lines. The final goal of this project would be to incorporate this reprojected information into training pipeline to get numbers better than the third line.

My second experiment is training a small network (1x1 convolution) on the groundtruth reprojected masks from previous frames. I use one-hot encoding, so each training input has shape $(25 * n)$xheightxwidth, where 25 is the number of categories (plus 1 for back ground) and n is the number of previous masks we passed in the network. The first version I trained has the masks from $t, t-1, t-2, t-3$. As one would expect, the model learned to take the groundtruth mask from t directly as output, and achieved perfect accuracy. **??** shows the weight that the model learned. The x-axis is the the input channels and the y-axis is the output channels. The model correctly learns to use the main diagonal primarily, correponding to using the groundtruth mask from the frame t. In the next version, I only passed in groundtruth masks from $t-1, t-2, t-3$ as input. As shown in **??**, the weights are largest in the three diagonals as one would expect. Out of the three diagonals, the main diagonal is the largest. It is consistent with our expectation that the the most recent mask $(t-1)$ is the most valuable. But it is good to see that the model is also using mask from $t-2$ and $t-3$ frames to some extent. This furthur confirms our pressumption that the reprojected masks from previous frames would help with the performance of the model. I also explored how some hyper-parameters impact the training process. Finally, I launched a training with a 3x3 convolutional network with the same parameters. It did not perform too much better than the 1x1 one.

?? shows the quantative results of these sanity check experiments. The convolutional networks performs slightly better than directly using the groundtruth $t-1$ mask, which makes sense since that is the most recent mask. The plots **??** of the predicted mask also matches this.

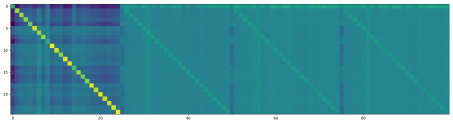

Figure 4.2: Weight that the model learned. Input is gt masks from t,t-1,t-2,t-3

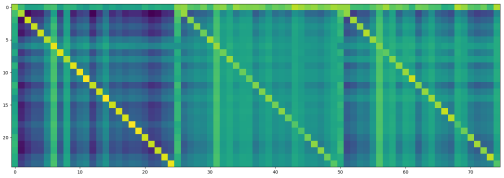

Figure 4.3: Weight that the model learned. Input is gt masks from t,t-1,t-2,t-3

4.5 EXPERIMENT WITH REPROJECTION

In the main experiment, I implemented the reprojection in feature space. As shown in **??**, the main part I added is the red part. In the dataloader, I added the depth map of the images and the camera matrices in addition to the images theselves. Then in the network, I added a layer that reproject the feature from one frame to another the same way I described in the previous section. In the implementation, I had to change the reprojection code a little to accomodate reprojection in the feature space. Since it is of a lower resolution, I needed to downsample the depth map. I tried a few difference modes of resizing to get the minimum amount of the aliasing effect. I found at the end the just using the nearest performs the best.

[TODO]

Figure 4.4: gt mask, t-1 mask, t-2 mask,t-3 mask and predicted mask. The prediction is from 1x1 conv network that is trained without gt

Method	SAIL-VOS class-specific AP	AP_{50}	AP^{P}_{50}	AP^{H}_{50}	AP^{L}_{50}	AP^{M}_{50}	AP^{S}_{50}
MaskAmodal	13.0	23.0	24.3	16.7	36.6	21.5	6.1
MaskJoint [**?**]	14.1	24.8	24.3	18.9	37.8	21.5	5.7
Best model (Base model)	**17.6**	**28.3**	**28.9**	**20.1**	**47.1**	**24.8**	**10.6**
Gt t-1 frame w/o reprojection	-	61.7	53.6	50.5	-	-	-
Gt t-1 frame w reprojection	**-**	**69.3**	**61.8**	**58.0**	**-**	**-**	**-**
Gt t-2 frame w/o reprojection	-	50.4	43.3	37.7	-	-	-
Gt t-2 frame w reprojection	**-**	**64.5**	**56.7**	**51.8**	**-**	**-**	**-**
Gt t-2 frame w reprojection	-	64.5	56.7	51.8	-	-	-
1x1 Conv w/ gt t,t-1,t-2, t-3 frames w reprojection	-	99.9	99.9	99.9	-	-	-
1x1 Conv w/ gt t-1,t-2, t-3 frames w reprojection	-	69.4	61.9	58.2	-	-	-
3x3 Conv w/ gt t-1,t-2, t-3 frames w reprojection	-	69.4	61.8	58.4	-	-	-

Table 4.1: Quantitative amodal segmentation results for the SAIL-VOS dataset using class-specific and class-agnostic settings.

CHAPTER 5: RESULT

Method	SAIL-VOS class-specific						
	AP	AP_{50}	AP^{P}_{50}	AP^{H}_{50}	AP^{L}_{50}	AP^{M}_{50}	AP^{S}_{50}
MaskAmodal	13.0	23.0	24.3	16.7	36.6	21.5	6.1
MaskJoint [**?**]	14.1	24.8	24.3	18.9	37.8	21.5	5.7
Base model(Amodal-Net)	**17.6**	**28.3**	**28.9**	**20.1**	**47.1**	**24.8**	**10.6**
Base model with reprojection	-	-	-	-	-	-	-
Base model without reprojection	-	-	-	-	-	-	-

Table 5.1: Quantitative amodal segmentation results for the SAIL-VOS dataset using class-specific and class-agnostic settings.

CHAPTER 6: DISCUSSION

CHAPTER 7: FUTURE WORK

CHAPTER 8: CONCLUSIONS

APPENDIX A: REPROJECTION CODE