Self-Supervised Scene De-occlusion

Xiaohang Zhan[1], Xingang Pan[1], Bo Dai[1], Ziwei Liu[1], Dahua Lin[1], and Chen Change Loy[2]

[1]CUHK - SenseTime Joint Lab, The Chinese University of Hong Kong
[2]Nanyang Technological University

[1]{zx017, px117, bdai, zwliu, dhlin}@ie.cuhk.edu.hk
[2]ccloy@ntu.edu.sg

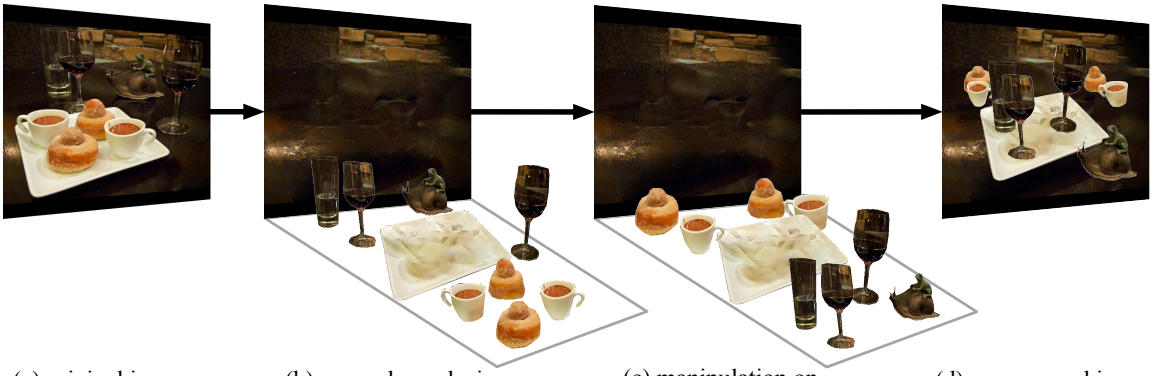


Figure 1: Scene de-occlusion decomposes an image, extracting cluttered objects in it into entities of indiviual intact objects. Orders and positions of the extracted objects can be manipulated to recompose new scenes.

Abstract

Natural scene understanding is a challenging task, particularly when encountering images of multiple objects that are partially occluded. This obstacle is given rise by varying object ordering and positioning. Existing scene understanding paradigms are able to parse only the visible parts, resulting in incomplete and unstructured scene interpretation. In this paper, we investigate the problem of scene de-occlusion, which aims to recover the underlying occlusion ordering and complete the invisible parts of occluded objects. We make the first attempt to address the problem through a novel and unified framework that recovers hidden scene structures without ordering and amodal annotations as supervisions. This is achieved via Partial Completion Network (PCNet)-mask (M) and -content (C), that learn to recover fractions of object masks and contents, respectively, in a self-supervised manner. Based on PCNet-M and PCNet-C, we devise a novel inference scheme to accomplish scene de-occlusion, via progressive ordering recovery, amodal completion and content completion. Extensive experiments on real-world scenes demonstrate the superior performance of our approach to other alternatives. Remarkably, our approach that is trained in a self-supervised manner achieves comparable results to fully-supervised methods. The proposed scene de-occlusion framework benefits many applications, including high-quality and controllable image manipulation and scene recomposition (see Fig. 1), as well as the conversion of existing modal mask annotations to amodal mask annotations. Project page: `https://xiaohangzhan.github.io/projects/deocclusion/`.

1. Introduction

Scene understanding is one of the foundations of machine perception. A real-world scene, regardless of its context, often comprises multiple objects of varying ordering and positioning, with one or more object(s) being occluded by other object(s). Hence, scene understanding systems should be able to process modal perception, *i.e.*, pars-

ing the directly visible regions, as well as amodal perception [1, 2, 3], *i.e.*, perceiving the intact structures of entities including invisible parts. The advent of advanced deep networks along with large-scale annotated datasets has facilitated many scene understanding tasks, *e.g.*, object detection [4, 5, 6, 7], scene parsing [8, 9, 10], and instance segmentation [11, 12, 13, 14]. Nonetheless, these tasks mainly concentrate on modal perception, while amodal perception remains rarely explored to date.

A key problem in amodal perception is *scene de-occlusion*, which involves the subtasks of recovering the underlying occlusion ordering and completing the invisible parts of occluded objects. While human vision system is capable of intuitively performing scene de-occlusion, elucidation of occlusions is highly challenging for machines. First, the relationships between an object that occludes other object(s), called an "occluder", and an object that is being occluded by other object(s), called an "occludee", is profoundly complicated. This is especially true when there are multiple "occluders" and "occludees" with high intricacies between them, namely an "occluder" that occludes multiple "occludees" and an "ocludee" that is occluded by multiple "occluders", forming a complex occlusion graph. Second, depending on the category, orientation, and position of objects, the boundaries of "occludee(s)" are elusive; no simple priors can be applied to recover the invisible boundaries.

A possible solution for scene de-occlusion is to train a model with *ground truth* of occlusion orderings and amodal masks (*i.e.*, intact instance masks). Such ground truth can be obtained either from synthetic data [15, 16] or from manual annotations on real-world data [17, 18, 19], each of which with specific limitations. The former introduces inevitable domain gap between the fabricated data used for training and the real-world scene in testing. The latter relies on subjective interpretation of individual annotators to demarcate occluded boundaries, therefore subjected to biases, and requires repeated annotations from different annotators to reduce noise, therefore are laborious and costly. A more practical and scalable way is to learn scene de-occlusion from the data itself rather than annotations.

In this work, we propose a novel self-supervised framework that tackles scene de-occlusion on real-world data *without manual annotations of occlusion ordering or amodal masks*. In the absence of ground truth, an end-to-end supervised learning framework is not applicable anymore. We therefore introduce a unique concept of *partial completion* of occluded objects. There are two core precepts in the partial completion notion that enables attainment of scene de-occlusion in a self-supervised manner. First, the process of completing an "occludee" occluded by multiple "occluders" can be broken down into a sequence of partial completions, with one "occluder" involved at a time. Second, the learning of making partial completion

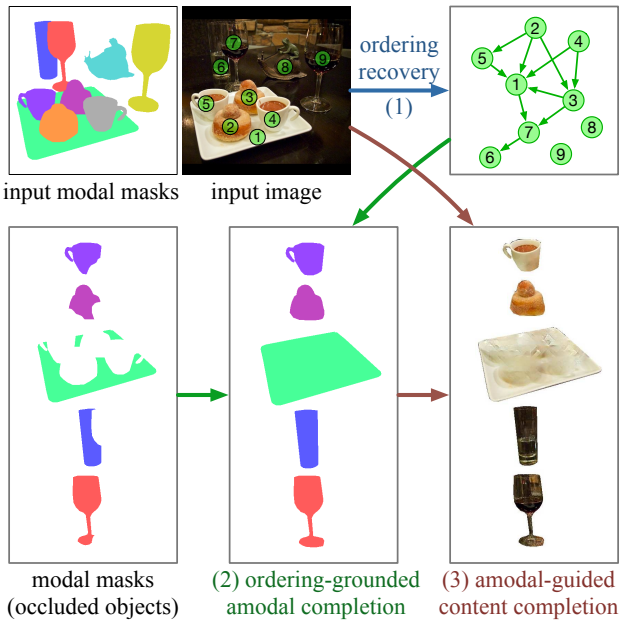


Figure 2: Given an input image and the associated modal masks, our framework solves scene de-occlusion progressively – 1) predicts occlusion ordering between different objects as a directed graph, 2) performs amodal completion grounded on the ordering graph, and 3) furnishes the occluded regions with content under the guidance of amodal predictions. The de-occlusion is achieved by two novel networks, PCNet-M and PCNet-C, which are trained without annotations of ordering or amodal masks.

can be achieved by further trimming down the "occludee" deliberately and training a network to recover the previous untrimmed occludee. We show that partial completion is sufficient to complete an occluded object progressively, as well as to facilitate the reasoning of occlusion ordering.

Partial completion is executed via two networks, *i.e.*, Partial Completion Network-mask and -content. We abbreviate them as PCNet-M and PCNet-C, respectively. PCNet-M is trained to *partially* recover the invisible mask of the "occludee" corresponding to an occluder, while PCNet-C is trained to *partially* fill in the recovered mask with RGB content. PCNet-M and PCNet-C form the two core components of our framework to address scene de-occlusion.

As illustrated in Fig. 2, the proposed framework takes a real-world scene and its corresponding modal masks of objects, derived from either annotations or predictions of existing modal segmentation techniques, as inputs. Our framework then streamlines three subtasks to be tackled progressively: **1) Ordering Recovery.** Given a pair of neighboring objects in which one can be occluding the other, following the principle that PCNet-M partially completes the mask of the "occludee" while keeping the "occluder" unmodified, the roles of the two objects are determined. We recover the ordering of all neighboring pairs

and obtain a directed graph that captures the occlusion order among all objects. **2) Amodal Completion.** For a specific "occludee", the ordering graph indicates all its "occluders". Grounded on this information and reusing PCNet-M, an amodal completion method is devised to fully complete the modal mask into an amodal mask of the "occludee". **3) Content Completion.** The predicted amodal mask indicates the occluded region of an "occludee". Using PCNet-C, we furnish RGB content into the invisible region. With such a progressive framework, we decompose a complicated scene into isolated and intact objects, along with a highly accurate occlusion ordering graph, allowing subsequent manipulation on the ordering and positioning of objects to recompose a new scene, as shown in Fig. 1.

We summarize our contributions as follows: **1)** We streamline scene de-occlusion into three subtasks, namely ordering recovery, amodal completion, and content completion. **2)** We propose PCNets and a novel inference scheme to perform scene de-occlusion without the need for corresponding manual annotations. Yet, we observe comparable results to fully-supervised approaches on datasets of real scenes. **3)** The self-supervised nature of our approach shows its potential to endow large-scale instance segmentation datasets, *e.g*., KITTI [20], COCO [21], *etc*., with high-accuracy ordering and amodal annotations. **4)** Our scene de-occlusion framework represents a novel enabling technology for real-world scene manipulation and recomposition, providing a new dimension for image editing.

2. Related Work

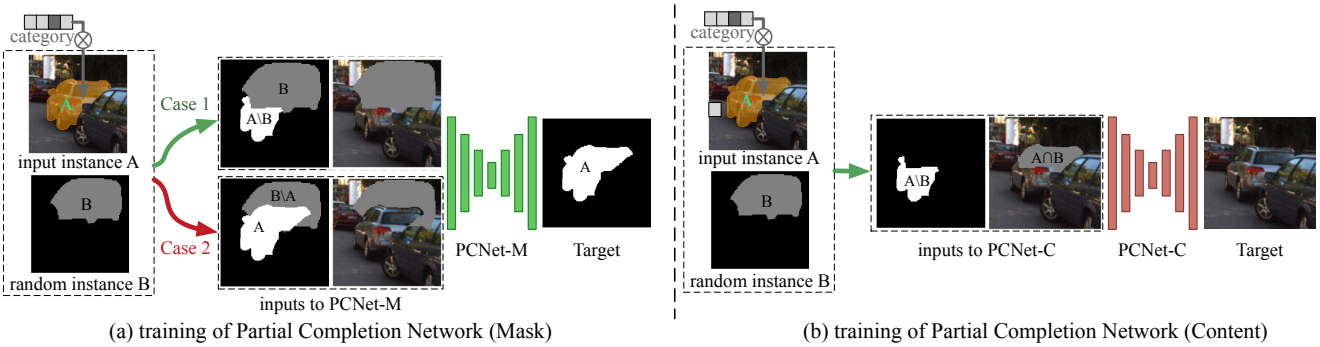
Ordering Recovery. In the unsupervised stream, Wu *et al*. [22] propose to recover ordering by re-composing the scene with object templates. However, they only demonstrate the system on toy data. Tighe *et al*. [23] build a prior occlusion matrix between classes on the training set and minimize quadratic programming to recover the ordering in testing. The inter-class occlusion prior ignores the complexity of realistic scenes. Other works [24, 25] rely on additional depth cues. However, depth is not reliable in occlusion reasoning, *e.g*., there is no depth difference if a piece of paper lies on a table. The assumption made by these works that farther objects are occluded by close ones also does not always hold. For example, as shown in Fig. 2. The plate (#1) is occluded by the coffee cup (#5), while the cup is farther in depth. In the supervised stream, several works manually annotate occlusion ordering [17, 18] or rely on synthetic data [16] to learn the ordering in a fully-supervised manner. Another stream of works on panoptic segmentation [26, 27] design end-to-end training procedures to resolve overlapping segments. However, they do not explicitly recover the full scene ordering.

Amodal Instance Segmentation. Modal segmentation, such as semantic segmentation [9, 10] and instance segmentation [11, 12, 13], aims at assigning categorical or object labels to visible pixels. Existing approaches for modal segmentation are not able to solve the de-occlusion problem. Different from modal segmentation, amodal instance segmentation aims at detecting objects as well as recovering the amodal (integrated) masks of them. Li *et al*. [28] produces dummy supervision through pasting artificial occluders, while the absence of explicit ordering increases the difficulty when complicated occlusion relationship is present. Other works take a fully-supervised learning approach by using either manual annotations [17, 18, 19] or synthetic data [16]. As mentioned above, it is costly and inaccurate to annotate invisible masks manually. Approaches relying on synthetic data are also confronted with domain gap issues. On the contrary, our approach can convert modal masks into amodal masks in a self-supervised manner. This unique ability facilitates the training of amodal instance segmentation networks without manual amodal annotations.

Amodal Completion. Amodal completion is slightly different from amodal instance segmentation. In amodal completion, modal masks are given at test time and the task is to complete the modal masks into amodal masks. Previous works on amodal completion typically rely on heuristic assumptions on the invisible boundaries to perform amodal completion with given ordering relationships. Kimia *et al*. [29] propose to adopt Euler Spiral in amodal completion. Lin *et al*. [30] use cubic Bézier curves. Silberman *et al*. [31] apply curve primitives including straight lines and parabolas. Since these studies still require ordering as the input, they cannot be adopted directly to solve de-occlusion problem. Besides, these unsupervised approaches mainly focus on toy examples with simple shapes. Kar *et al*. [32] use keypoint annotations to align 3D object templates to 2D image objects, so as to generate the ground truth of amodal bounding boxes. Ehsani *et al*. [15] leverage 3D synthetic data to train an end-to-end amodal completion network. Similar to unsupervised methods, our framework does not need annotations of amodal masks or any kind of 3D/synthetic data. In contrast, our approach is able to solve amodal completion in highly cluttered natural scenes, whereas other unsupervised methods fall short.

3. Our Scene De-occlusion Approach

The proposed framework aims at **1)** recovering occlusion ordering and **2)** completing amodal masks and content of occluded objects. To cope with the absence of manual annotations of occlusion ordering and amodal masks, we design a way to train the proposed PCNet-M and PCNet-C to complete instances partially in a self-supervised manner. With the trained networks, we further propose a progressive inference scheme to perform ordering recovery, ordering-grounded amodal completion, and amodal-constrained content completion to complete objects.



(a) training of Partial Completion Network (Mask)

(b) training of Partial Completion Network (Content)

Figure 3: The training procedure of the PCNet-M and the PCNet-C. Given an instance A as the input, we randomly sample another instance B from the whole dataset and position it randomly. Note that we only have modal masks of both A and B. (a) PCNet-M is trained by switching two cases. Case 1 (A erased by B) follows the partial completion mechanism where PCNet-M is encouraged to partially complete A. Case 2 prevents PCNet-M from over completing A. (b) PCNet-C uses $A \cap B$ to erase A and learn to fill in the RGB content of the erased region. It also takes in $A \backslash B$ as an additional input. The modal mask of A is multiplied with its category id if available.

3.1. Partial Completion Networks (PCNets)

Given an image, it is easy to obtain the modal masks of objects via off-the-shelf instance segmentation frameworks. However, their amodal masks are unavailable. Even worse, we do not know whether these modal masks are intact, making the learning of full completion of an occluded instance extremely challenging. The problem motivates us to explore self-supervised partial completion.

Motivation. Suppose an instance's modal mask constitutes a pixel set M, we denote the ground truth amodal mask as G. Supervised approaches solve the full completion problem of $M \xrightarrow{f_\theta} G$, where f_θ denotes the full completion model. This full completion process can be broken down into a sequence of partial completions $M \xrightarrow{p_\theta} M_1 \xrightarrow{p_\theta} M_2 \xrightarrow{p_\theta} \cdots \xrightarrow{p_\theta} G$ if the instance is occluded by multiple "occluders", where M_k is the intermediate states, p_θ denotes the partial completion model.

Since we still do not have any ground truth to train the partial completion step p_θ, we take a step back by further trimming down M randomly to obtain M_{-1} s.t. $M_{-1} \subset M$. Then we train p_θ via $M_{-1} \xrightarrow{p_\theta} M$. The self-supervised partial completion approximates the supervised one, laying the foundation of our PCNets. Based on such a self-supervised notion, we introduce Partial Completion Networks (PCNets). They contain two networks, respectively, for mask (PCNet-M) and content completion (PCNet-C).

PCNet-M for Mask Completion. The training of PCNet-M is shown in Fig. 3 (a). We first prepare the training data. Given an instance A along with its modal mask M_A from the dataset D with instance-level annotations, we randomly sample another instance B from D and position it randomly to acquire a mask M_B. Here we regard M_A and M_B as sets of pixels. There are two input cases, in which different input is fed to the network:

1) The *first case* corresponds to the aforementioned partial completion strategy. We define M_B as an eraser, and use B to erase part of A to obtain $M_{A\backslash B}$. In this case, the PCNet-M is trained to recover the original modal mask M_A from $M_{A\backslash B}$, conditioned on M_B.

2) The *second case* serves as a regularization to discourage the network from over-completing an instance if the instance is not occluded. Specifically, $M_{B\backslash A}$ that does not invade A is regarded as the eraser. In this case, we encourage the PCNet-M to retain the original modal mask M_A, conditioned on $M_{B\backslash A}$. Without case 2, the PCNet-M always encourage increment of pixels, which may result in over-completion of an instance if it is not occluded by other neighboring instances.

In both cases, the erased image patch serves as an auxiliary input. We formulate the loss functions as follows:

$$
\begin{aligned}
L_1 &= \frac{1}{N} \sum_{A,B \in D} L\left(P_\theta^{(\mathrm{m})}\left(M_{A\backslash B}\,; M_B, I\backslash M_B\right), M_A\right), \\
L_2 &= \frac{1}{N} \sum_{A,B \in D} L\left(P_\theta^{(\mathrm{m})}\left(M_A\,; M_{B\backslash A}, I\backslash M_{B\backslash A}\right), M_A\right),
\end{aligned} \tag{1}
$$

where $P_\theta^{(\mathrm{m})}(\star)$ is our PCNet-M network, θ represents the parameters to optimize, I is the image patch, L is Binary Cross-Entropy Loss. We formulate the final loss function as $L^{(\mathrm{m})} = xL_1 + (1-x)L_2, x \sim Bernoulli(\gamma)$, where γ is the probability to choose case 1. The random switching between the two cases forces the network to understand the ordering relationship between the two neighboring instances from their shapes and border, so as to determine whether to complete the instance or not.

PCNet-C for Content Completion. PCNet-C follows a similar intuition of PCNet-M, while the target to complete is RGB content. As shown in Fig. 3 (b), the input instances A and B are the same as that for PCNet-M. Image pixels in region $M_{A\cap B}$ are erased, and PCNet-C aims at predicting

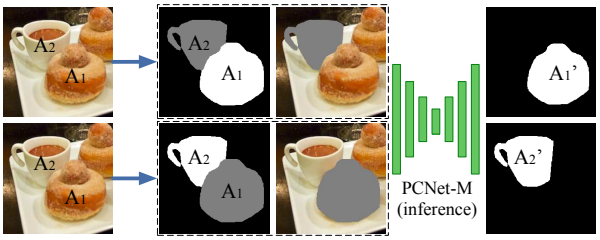


Figure 4: Dual-Completion for ordering recovery. To recover the ordering between a pair of neighboring instances A_1 and A_2, we switch the role of the target object (in white) and the eraser (in gray). The increment of A_2 is larger than that of A_1, thus A_2 is identified as the "occludee".

the missing content. Besides, PCNet-C also takes in the remaining mask of A, *i.e.*, $M_{A\setminus B}$ to indicate that it is A rather than other objects, that is painted. Hence, it cannot be simply replaced by standard image inpainting approaches. The loss of PCNet-C to minimize is formulated as follows:

$$L^{(\mathrm{c})} = \frac{1}{N} \sum_{A,B \in D} L\left(P_\theta^{(\mathrm{c})}\left(I \backslash M_{A \cap B}; M_{A\setminus B}, M_{A\cap B}\right), I\right), \tag{2}$$

where $P_\theta^{(\mathrm{c})}$ is our PCNet-C network, I is the image patch, L represents the loss function consisting of common losses in image inpainting including l_1, perceptual and adversarial loss. Similar to PCNet-M, the training of PCNet-C via learning partial completion enables full completion of the instance content at test time.

3.2. Dual-Completion for Ordering Recovery

The target ordering graph is composed of pair-wise occlusion relationships between all neighboring instance pairs. A neighboring instance pair is defined as two instances whose modal masks are connected, thus one of them possibly occludes the other. As shown in Fig. 4, given a pair of neighboring instances A_1 and A_2, we first regard A_1's modal mask M_{A_1} as the target to complete. M_{A_2} serves as the eraser to obtain the increment of A_1, *i.e.*, $\Delta_{A_1|A_2}$. Symmetrically, we also obtain the increment of A_2 conditioned on A_1, *i.e.*, $\Delta_{A_2|A_1}$. The instance gaining a larger increment in partial completion is supposed to be the "occludee". Hence, we infer the order between A_1 and A_2 via comparing their incremental area, as follows:

$$\begin{aligned}
\Delta_{A_1|A_2} &= P_\theta^{(\mathrm{m})}\left(M_{A_1}\,;\, M_{A_2}, I\backslash M_{A_2}\right) \setminus M_{A_1}, \\
\Delta_{A_2|A_1} &= P_\theta^{(\mathrm{m})}\left(M_{A_2}\,;\, M_{A_1}, I\backslash M_{A_1}\right) \setminus M_{A_2}, \\
O\left(A_1, A_2\right) &= \begin{cases} 0, & \text{if } |\Delta_{A_1|A_2}| = |\Delta_{A_2|A_1}| = 0 \\ 1, & \text{if } |\Delta_{A_1|A_2}| < |\Delta_{A_2|A_1}| \\ \text{-1}, & \text{otherwise} \end{cases},
\end{aligned} \tag{3}$$

where $O\left(A_1, A_2\right) = 1$ indicates that A_1 occludes A_2. If A_1 and A_2 are not neighboring, $O\left(A_1, A_2\right) = 0$. Note that

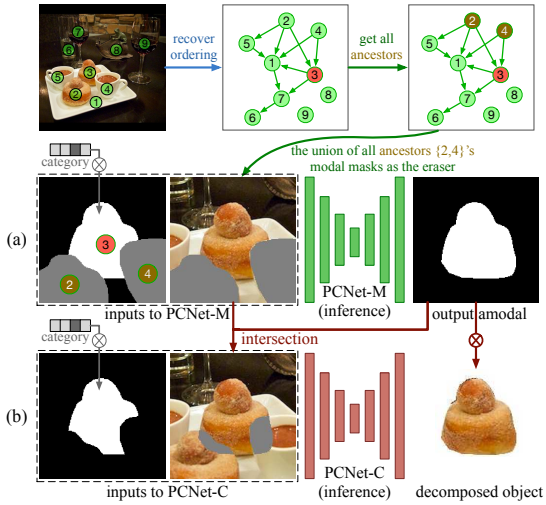


Figure 5: (a) Ordering-grounded amodal completion takes the modal mask of the target object (#3) and all its ancestors (#2, #4), as well as the erased image as inputs. With the trained PCNet-M, it predicts the amodal mask of object #3. (b) The intersection of the amodal mask and the ancestors indicates the invisible region of object #3. Amodal-constrained content completion (red arrows) adopts the PCNet-C to fill in the content in the invisible region.

in practice the probability of $\left|\Delta_{A_1|A_2}\right| = \left|\Delta_{A_2|A_1}\right| > 0$ is zero, thus does not need to be specifically considered here. Performing Dual-Completion for all neighboring pairs provides us the scene occlusion ordering, which can be represented as a directed graph as shown in Fig. 2. The nodes in the graph represent objects, while edges indicate the directions of occlusion between neighboring objects. Note that it is not necessarily to be acyclic, as shown in Fig. 7.

3.3. Amodal and Content Completion

Ordering-Grounded Amodal Completion. We can perform ordering-grounded amodal completion after estimating the ordering graph. Suppose we need to complete an instance A, we first find all ancestors of A in the graph as the "occluders" of this instance via breadth-first searching (BFS). Since the graph is not necessarily to be acyclic, we adapt the BFS algorithm accordingly. Interestingly, we find that the trained PCNet-M is generalizable to use the union of all ancestors as the eraser. Hence, we do not need to iterate the ancestors and apply PCNet-M to partially complete A step by step. Instead, we perform amodal completion in one step conditioned on the union of all ancestors' modal masks. Denoting the ancestors of A as $\{\mathrm{anc}_i^A, i = 1, 2, \cdots, k\}$, we perform amodal completion as follows:

$$\begin{aligned}
Am_A &= P_\theta^{(\mathrm{m})}\left(M_A\,;\, M_{\mathrm{anc}^A}, I\backslash M_{\mathrm{anc}^A}\right), \\
M_{\mathrm{anc}^A} &= \bigcup_{i=1}^{k} M_{\mathrm{anc}_i^A},
\end{aligned} \tag{4}$$

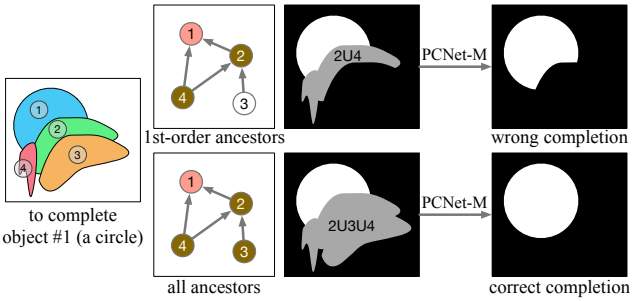


Figure 6: This figure shows why we need to find all ancestors rather than only the first-order ancestors, though higher-order ancestors do not directly occlude this instance. Higher-order ancestors (*e.g.*, instance #3) may indirectly occlude the target instance (#1), thus need to be taken into account.

where Am_A is the result of amodal mask, $M_{\text{anc}_i^A}$ is the modal mask of i-th ancestor. An example is shown in Fig. 5 (a). Fig. 6 shows the reason we use all ancestors rather than only the first-order ancestor.

Amodal-Constrained Content Completion. In previous steps, we obtain the occlusion ordering graph and the predicted amodal mask of each instance. Next, we complete the occluded content of them. As shown in Fig. 5 (b), the intersection of predicted amodal mask and the ancestors $Am_A \cap M_{\text{anc}^A}$ indicates the missing part of A, regarded as the eraser for PCNet-C. Then we apply a trained PCNet-C to fill in the content as follows:

$$\begin{aligned} C_A &= P_\theta^{(c)}\left(I \backslash M_E\ ;\ M_A, M_E\right) \circ Am_A, \\ M_E &= Am_A \cap M_{\text{anc}^A}, \end{aligned} \tag{5}$$

where C_A is the decomposed content of A from the scene. For background contents, we use the union of all foreground instances as the eraser. Different from image inpainting that is unaware of occlusion, content completion is performed on the estimated occluded regions.

4. Experiments

We now evaluate our method in various applications including *ordering recovery*, *amodal completion*, *amodal instance segmentation*, and *scene manipulation*. The implementation details and more qualitative results can be found in the supplementary materials.

Datasets. 1) KINS [18], originated from KITTI [20], is a large-scale traffic dataset with annotated modal and amodal masks of instances. PCNets are trained on the training split (7,474 images, 95,311 instances) with modal annotations. We test our de-occlusion framework on the testing split (7,517 images, 92,492 instances). **2) COCOA** [17] is a subset of COCO2014 [21] while annotated with pair-wise ordering, modal, and amodal masks. We train PCNets on the training split (2,500 images, 22,163 instances) using modal annotations and test on the validation split (1,323 images, 12,753 instances). The categories of instance are unavailable for this dataset. Hence, we set the category id constantly as 1 in training PCNets for this dataset.

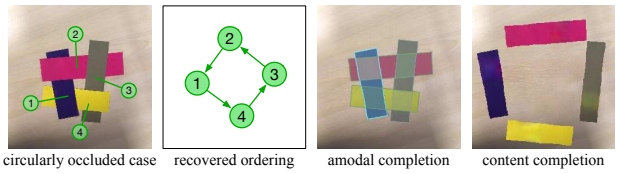


Figure 7: Our framework is able to solve circularly occluded cases. Since such case is rare, we cut four pieces of paper to compose it.

Table 1: Ordering estimation on COCOA validation and KINS testing sets, reported with pair-wise accuracy on occluded instance pairs.

method	gt order (train)	COCOA	KINS
Supervised			
OrderNet^{M} [17]	✔	81.7	87.5
OrderNet^{M+I} [17]	✔	88.3	94.1
Unsupervised			
Area	✘	62.4	77.4
Y-axis	✘	58.7	81.9
Convex	✘	76.0	76.3
Ours	✘	**87.1**	**92.5**

4.1. Comparison Results

Ordering Recovery. We report ordering recovery performance on COCOA and KINS in Table 1. We reproduced the OrderNet proposed in [17] to obtain the supervised results. Baselines include sorting bordered instance pairs by *Area*[1], *Y-axis* (instance closer to image bottom in front), and *Convex* prior. For baseline *Convex*, we compute convex hull on modal masks to approximate amodal completion, and the object with more increments is regarded as the occludee. All baselines have been adjusted to achieve their respective best performances. On both benchmarks, our method achieves much higher accuracies than baselines, comparable to the supervised counterparts. An interesting case is shown in Fig. 7, where four objects are circularly overlapped. Since our ordering recovery algorithm recovers pair-wise ordering rather than sequential ordering, it is able to solve this case and recover the cyclic directed graph.

Amodal Completion. We first introduce the baselines. For the *supervised* method, amodal annotation is available. A UNet is trained to predict amodal masks from modal masks end-to-end. *Raw* means no completion is performed. *Convex* represents computing the convex hull of the modal mask

[1]We optimize this heuristic depending on each dataset – a larger instance is treated as a front object for KINS, and opposite for COCOA.

Table 2: Amodal completion on COCOA validation and KINS testing sets, using ground truth modal masks.

method	amodal (train)	COCOA %mIoU	KINS %mIoU
Supervised	✔	82.53	94.81
Raw	✘	65.47	87.03
Convex^R	✘	74.43	90.75
Ours (NOG)	✘	76.91	93.42
Ours (OG)	✘	**81.35**	**94.76**

Table 3: Amodal completion on KINS testing set, using predicted modal masks (mAP 52.7%).

method	amodal (train)	KINS %mIoU
Supervised	✔	87.29
Raw	✘	82.05
Convex^R	✘	84.12
Ours (NOG)	✘	85.39
Ours (OG)	✘	**86.26**

as the amodal mask. Since the convex hull usually leads to over-completion, *i.e.*, extending the visible mask, we improve this baseline by using predicted order to refine the convex hull, constituting a stronger baseline: *Convex^R*. It performs pretty well for naturally convex objects. *Ours (NOG)* represents the non-ordering-grounded amodal completion that relies on our PCNet-M and regards all neighboring objects as the eraser rather than using occlusion ordering to search the ancestors. *Ours (OG)* is our ordering-grounded amodal completion method.

We evaluate amodal completion on ground truth modal masks, as shown in Table 2. Our method surpasses the baseline approaches and are comparable to the supervised counterpart. The comparison between *OG* and *NOG* shows the importance of ordering in amodal completion. As shown in Fig. 9, some of our results are potentially more natural than manual annotations.

Apart from using ground truth modal masks as the input in testing, we also verify the effectiveness of our approach with predicted modal masks as the input. Specifically, we train a UNet to predict modal masks from an image. In order to correctly match the modal and the corresponding ground truth amodal masks in evaluation, we use the bounding box as an additional input to this network. We predict the modal masks on the testing set, yielding 52.7% mAP to the ground truth modal masks. We use the predicted modal masks as the input to perform amodal completion. As shown in Table 3, our approach still achieves high performance, comparable to the supervised counterpart.

Label Conversation for Amodal instance segmentation. Amodal instance segmentation aims at detecting instances and predicting amodal masks from images simultaneously.

Table 4: Amodal instance segmentation on KINS testing set. Convex^R means using predicted order to refine the convex hull. In this experimental setting, all methods detect and segment instances from raw images. Hence, modal masks are not used in testing.

Ann. source	modal (train)	amodal (train)	%mAP
GT [18]	✘	✔	29.3
Raw	✔	✘	22.7
Convex	✔	✘	22.2
Convex^R	✔	✘	25.9
Ours	✔	✘	**29.3**

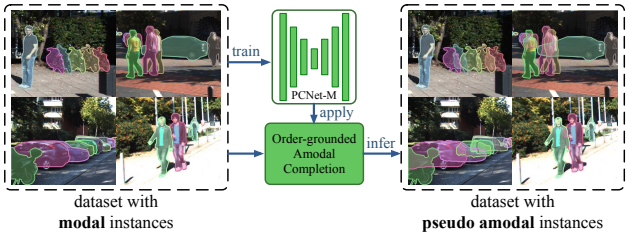


Figure 8: By training the self-supervised PCNet-M on a modal dataset (*e.g.*, KITTI shown here) and applying our amodal completion algorithm on the same dataset, we are able to freely convert modal annotations into pseudo amodal annotations. Note that such self-supervised conversion is intrinsically different from training a supervised model on a small labeled amodal dataset and applying it to a larger modal dataset, where the generalizability between different datasets can be an issue.

With our approach, one can convert an existing dataset with modal annotations into the one with *pseudo amodal annotations*, thus allowing amodal instance segmentation network training without manual amodal annotations. This is achieved by training PCNet-M on the modal mask training split, and applying our amodal completion algorithm on the same training split to obtain the corresponding amodal masks, as shown in Fig. 8, To evaluate the quality of the pseudo amodal annotations, we train a standard Mask R-CNN [12] for amodal instance segmentation following the setting in [18]. All baselines follow the same training protocol, except that the amodal annotations for training are different. As shown in Table 4, using our inferred amodal bounding boxes and masks, we achieve the same performance (mAP 29.3%) as the one using manual amodal annotations. Besides, our inferred amodal masks in the training set are highly consistent with the manual annotations (mIoU 95.22%). The results suggest a high applicability of our method for obtaining reliable pseudo amodal mask annotations, relieving burdens of manual annotation on large-scale instance-level datasets.

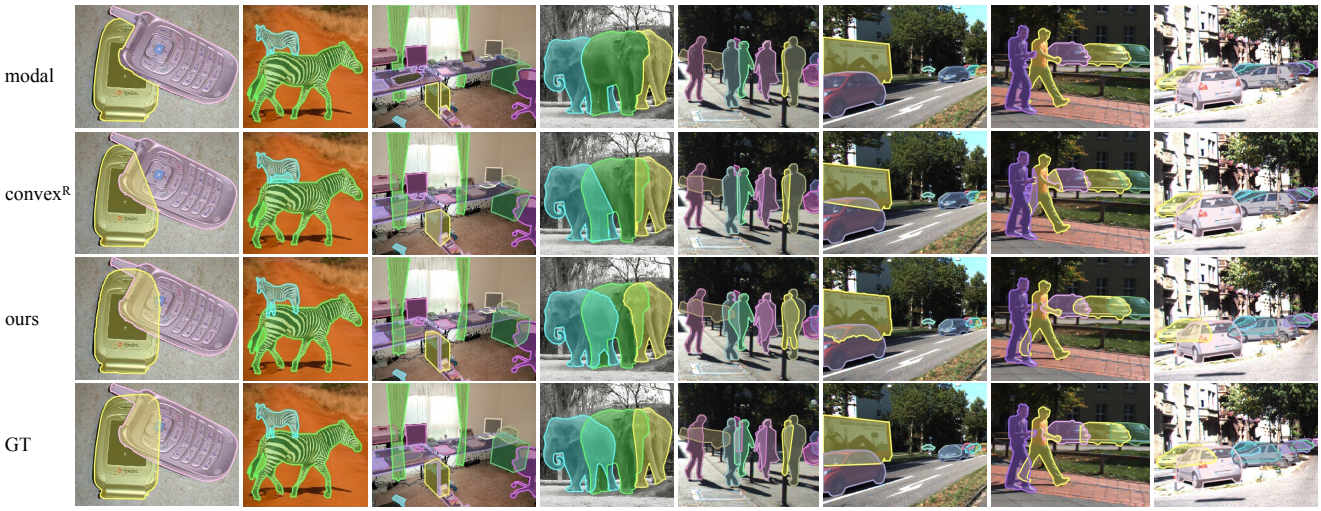


Figure 9: Amodal completion results. Our results are potentially more natural than manual annotations (GT) in some cases, especially for instances in yellow.

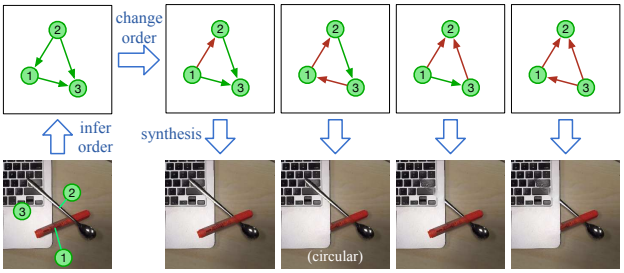


Figure 10: Scene synthesis by changing the ordering graph. Reversed orderings are shown in red arrows. Uncommon cases with circular ordering can also be synthesized.

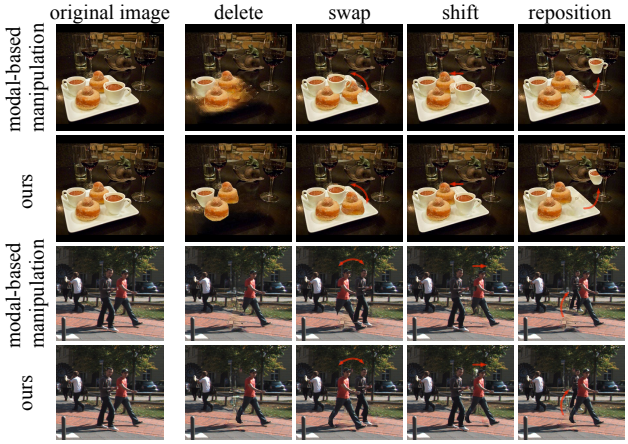


Figure 11: This figure shows rich and high-quality manipulations, including deleting, swapping, shifting and repositioning instances, enabled by our approach. The baseline method *modal-based manipulation* is based on image inpainting, where modal masks are provided, order and amodal masks are unknown. Better in zoomed-in view. More examples can be found in the supplementary material.

4.2. Application on Scene Manipulation

Our scene de-occlusion framework allows us to decompose a scene into the background and isolated completed objects, along with an occlusion ordering graph. Therefore, manipulating scenes by controlling order and positions is made possible. Fig. 10 shows scene synthesis by controlling order only. Fig. 11 shows more manipulation cases, indicating that our de-occlusion framework, though trained without any extra information compared to the baseline, enables high-quality *occlusion-aware manipulation*.

5. Conclusion

To summarize, we have proposed a unified scene de-occlusion framework equipped with self-supervised PC-Nets trained without ordering or amodal annotations. The framework is applied in a progressive way to recover occlusion orderings, then perform amodal and content completion. It achieves comparable performances to the fully-supervised counterparts on real-world datasets. It is applicable to convert existing modal annotations to amodal annotations. Quantitative results show their equivalent efficacy to manual annotations. Furthermore, our framework enables high-quality occlusion-aware scene manipulation, providing a new dimension for image editing.

Acknowledgement: This work is supported by the SenseTime-NTU Collaboration Project, Collaborative Research grant from SenseTime Group (CUHK Agreement No. TS1610626 & No. TS1712093), Singapore MOE AcRF Tier 1 (2018-T1-002-056), NTU SUG, and NTU NAP.

References

[1] Gaetano Kanizsa. *Organization in vision: Essays on Gestalt perception*. Praeger Publishers, 1979.

[2] Stephen E Palmer. *Vision science: Photons to phenomenology*. MIT press, 1999.

[3] Steven Lehar. Gestalt isomorphism and the quantification of spatial perception. *Gestalt theory*, 21:122–139, 1999.

[4] Pedro F Felzenszwalb, Ross B Girshick, and David McAllester. Cascade object detection with deformable part models. In *CVPR*, pages 2241–2248. IEEE, 2010.

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[6] Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. In *CVPR*, 2019.

[7] Kai Chen, Jiaqi Wang, Shuo Yang, Xingcheng Zhang, Yuanjun Xiong, Chen Change Loy, and Dahua Lin. Optimizing video object detection via a scale-time lattice. In *CVPR*, June 2018.

[8] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015.

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *PAMI*, 40(4):834–848, 2017.

[10] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017.

[11] Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In *ECCV*, pages 534–549. Springer, 2016.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[13] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *CVPR*, pages 4974–4983, 2019.

[14] Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. Carafe: Content-aware reassembly of features. In *ICCV*, October 2019.

[15] Kiana Ehsani, Roozbeh Mottaghi, and Ali Farhadi. Segan: Segmenting and generating the invisible. In *CVPR*, pages 6144–6153, 2018.

[16] Yuan-Ting Hu, Hong-Shuo Chen, Kexin Hui, Jia-Bin Huang, and Alexander G Schwing. Sail-vos: Semantic amodal instance level video object segmentation-a synthetic dataset and baselines. In *CVPR*, pages 3105–3115, 2019.

[17] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *CVPR*, pages 1464–1472, 2017.

[18] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. In *CVPR*, pages 3014–3023, 2019.

[19] Patrick Follmann, Rebecca Kö Nig, Philipp Hä Rtinger, Michael Klostermann, and Tobias Bö Ttger. Learning to see the invisible: End-to-end trainable amodal instance segmentation. In *WACV*, pages 1328–1336. IEEE, 2019.

[20] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, and Piotr Dollár. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[22] Jiajun Wu, Joshua B Tenenbaum, and Pushmeet Kohli. Neural scene de-rendering. In *CVPR*, 2017.

[23] Joseph Tighe, Marc Niethammer, and Svetlana Lazebnik. Scene parsing with object instances and occlusion ordering. In *CVPR*, pages 3748–3755, 2014.

[24] Derek Hoiem, Andrew N Stein, Alexei A Efros, and Martial Hebert. Recovering occlusion boundaries from a single image. In *ICCV*, 2007.

[25] Pulak Purkait, Christopher Zach, and Ian Reid. Seeing behind things: Extending semantic segmentation to occluded regions. *arXiv preprint arXiv:1906.02885*, 2019.

[26] Huanyu Liu, Chao Peng, Changqian Yu, Jingbo Wang, Xu Liu, Gang Yu, and Wei Jiang. An end-to-end network for panoptic segmentation. In *CVPR*, pages 6172–6181, 2019.

[27] Justin Lazarow, Kwonjoon Lee, and Zhuowen Tu. Learning instance occlusion for panoptic segmentation. *arXiv preprint arXiv:1906.05896*, 2019.

[28] Ke Li and Jitendra Malik. Amodal instance segmentation. In *ECCV*, pages 677–693. Springer, 2016.

[29] Benjamin B Kimia, Ilana Frankel, and Ana-Maria Popescu. Euler spiral for shape completion. *IJCV*, 54(1-3):159–182, 2003.

[30] Hongwei Lin, Zihao Wang, Panpan Feng, Xingjiang Lu, and Jinhui Yu. A computational model of topological and geometric recovery for visual curve completion. *Computational Visual Media*, 2(4):329–342, 2016.

[31] Nathan Silberman, Lior Shapira, Ran Gal, and Pushmeet Kohli. A contour completion model for augmenting surface reconstructions. In *ECCV*, 2014.

[32] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Amodal completion and size constancy in natural scenes. In *ICCV*, pages 127–135, 2015.