3D Scene Graph: A structure for unified semantics, 3D space, and camera

Iro Armeni[1]   Zhi-Yang He[1]   JunYoung Gwak[1]   Amir R. Zamir[1,2]

Martin Fischer[1]   Jitendra Malik[2]   Silvio Savarese[1]

[1] Stanford University   [2] University of California, Berkeley

http://3dscenegraph.stanford.edu

Abstract

A comprehensive semantic understanding of a scene is important for many applications - but in what space should diverse semantic information (e.g., objects, scene categories, material types, 3D shapes, etc.) be grounded and what should be its structure? Aspiring to have one unified structure that hosts diverse types of semantics, we follow the Scene Graph paradigm in 3D, generating a 3D Scene Graph. Given a 3D mesh and registered panoramic images, we construct a graph that spans the entire building and includes semantics on objects (e.g., class, material, shape and other attributes), rooms (e.g., function, illumination type, etc.) and cameras (e.g., location, etc.), as well as the relationships among these entities.

However, this process is prohibitively labor heavy if done manually. To alleviate this we devise a semi-automatic framework that employs existing detection methods and enhances them using two main constraints: I. framing of query images sampled on panoramas to maximize the performance of 2D detectors, and II. multi-view consistency enforcement across 2D detections that originate in different camera locations.

1. Introduction

Where should semantic information be grounded and what structure should it have to be most useful and invariant? This is a fundamental question for a content that preoccupies a number of domains, such as Computer Vision and Robotics. There is a clear number of components in play: geometry of the objects and space, categories of the entities therein, and the viewpoint from which the scene is being observed (i.e. the camera pose).

On the *space* where this information can be grounded, the most commonly employed choice is images. However, the use of images for this purpose is not ideal since it presents a variety of weaknesses, such as pixels being
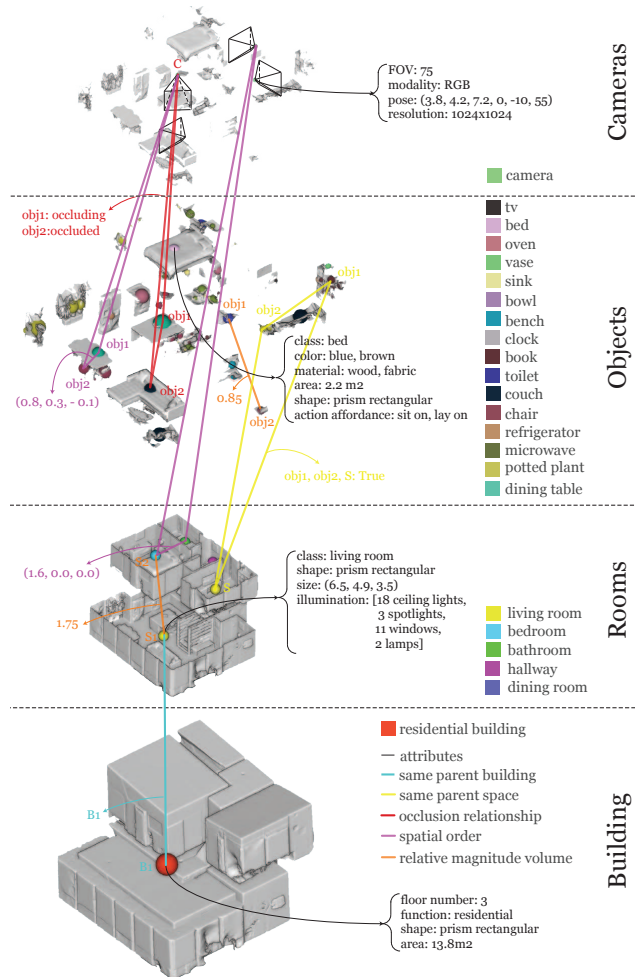

Figure 1. **3D Scene Graph:** It consists of 4 layers, that represent semantics, 3D space and camera. Elements are nodes in the graph and have certain attributes. Edges are formed between them to denote relationships (e.g., occlusion, relative volume, etc.).

highly variant to any parameter change, the absence of an object's entire geometry, and more. An ideal space for this purpose would be at minimum (a) invariant to as many

changes as possible, and (b) easily and deterministically connected to various output ports that different domains and tasks require, such as images or videos. To this end, we articulate that 3D space is more stable and invariant, yet connected to images and other pixel and non-pixel output domains (e.g. depth). Hence, we ground semantic information there, and project it to other desired spaces as needed (e.g., images, etc.). Specifically, this means that the information is grounded on the underlying 3D mesh of a building. This approach presents a number of useful values, such as free 3D, amodal, occlusion, and open space analysis. More importantly, semantics can be projected onto any number of visual observations (images and videos) which provides them with annotations without additional cost.

What should be the *structure*? Semantic repositories use different representations, such as object class and natural language captions. The idea of scene graph has several advantages over other representations that make it an ideal candidate. It has the ability to encompass more information than just object class (e.g., ImageNet [14]), yet it contains more structure and invariance than natural language captions (e.g., CLEVR [22]). We augment the basic scene graph structure, such as the one in Visual Genome [27], with essential 3D information and generate a *3D Scene Graph*.

We view 3D Scene Graph as a layered graph, with each level representing a different entity: building, room, object, and camera. More layers can be added that represent other sources of semantic information. Similar to the 2D scene graph, each entity is augmented with several attributes and gets connected to others to form different types of relationships. To construct the 3D Scene Graph, we combine state-of-the-art algorithms in a mainly automatic approach to semantic recognition. Beginning from 2D, we gradually aggregate information in 3D using two constraints: *framing* and *multi-view consistency*. Each constraint provides more robust final results and consistent semantic output.

The contributions of this paper can be summarized as:

- We extend the scene graph idea in [27] to 3D space and ground semantic information there. This gives free computation for various attributes and relationships.

- We propose a two-step robustification approach to optimizing semantic recognition using imperfect existing detectors, which allows the automation of a mainly manual task.

- We augment the Gibson Environment's [44] database with 3D Scene Graph as an additional modality and make it publicly available at 3dscene-graph.stanford.edu.

2. Related Work

Scene Graph A diverse and structured repository is Visual Genome [27], which consists of 2D images in the wild

of objects and people. Semantic information per image is encoded in the form of a scene graph. In addition to object class and location, it offers attributes and relationships. The nodes and edges in the graph stem from natural language captions that are manually defined. To address naming inconsistencies due to the free form of annotations, entries are canonicalized before getting converted into the final scene graph. In our work, semantic information is generated in an automated fashion - hence significantly more efficient, already standardized, and to a great extent free from human subjectivity. Although using predefined categories can be restrictive, it is compatible with current learning systems. In addition, 3D Scene Graph allows to compute from 3D an unlimited number of spatially consistent 2D scene graphs and provides numerically accurate quantification to relationships. However, our current setup is limited to indoor static scenes, hence not including outdoor related attributes or action-related relationships, like Visual Genome.

Using Scene Graphs Following Visual Genome, several works emerged that employ or generate scene graphs. Examples are on scene graph generation [30, 46], image captioning/description [26, 3, 23], image retrieval [24] and visual question-answering [17, 51]. Apart from vision-language tasks, there is also a focus relationship and action detection [34, 31, 47]. A 3D Scene Graph will similarly enable, in addition to common 3D vision tasks, others to emerge in the combination of 3D space, 2D-2.5D images, video streams, and language.

Utilizing Structure in Prediction Adding structure to prediction, usually in the form of a graph, has proven beneficial for several tasks. One common application is that of Conditional Random Fields (CRF) [28] for *semantic segmentation*, often used to provide globally smooth and consistent results to local predictions [43, 25]. In the case of *robot navigation*, employing semantic graphs to abstract the physical map allows the agent to learn by understanding the relationship between semantic nodes independent of the metric space, which results to easier generalization across spaces [42]. Graph structures are also commonly used in *human-object interaction* tasks [39] and other spatio-temporal problems [20], creating connections among nodes within and across consecutive video frames, hence extending structure to include, in addition to space, also time. Grammars that combine geometry, affordance and appearance have been used toward *holistic scene parsing* in images, where information about the scene and objects is captured in a hierarchical tree structure [11, 48, 21, 19]. Nodes represent scene or object components and attributes, whereas edges can represent decomposition (e.g., a scene into objects, etc.) or relationship (e.g., supporting, etc.). Similar to such works, our structure combines different semantic information. However, it can capture global 3D relationships on the building scale and provides greater free-

dom in the definition of the graph by placing elements in different layers. This removes the need for direct dependencies across them (e.g., between a scene type and object attributes). Another interesting example is that of Visual Memex [36] that leverages a graph structure to encode contextual and visual similarities between objects without the notion of categories, with the goal to predict the object class laying under a masked region. Zhu et al. [50] used a knowledge base representation for the task of object affordance reasoning that places edges between different nodes of objects, attributes, and affordances. These examples incorporate different types of semantic information in a unified structure for multi-modal reasoning. The above echoes the value of having richly structured information.

Semantic Databases Existing semantic repositories are fragmented to specific types of visual information, with their majority focusing on object class labels and spatial span/positional information (e.g., segmentation masks/bounding boxes). These can be further sub-grouped based on the visual modality (e.g., RGB, RGBD, point clouds, 3D mesh/CAD models, etc.) and content scene (e.g., indoor/outdoor, object only, etc.). Among them, a handful provides multimodal data grounded on 3D meshes (e.g., 2D-3D-S [6], Matterport3D [10]). The Gibson database, recently presented in [44], consists of several hundreds of 3D mesh models with registered panoramic images. It is approximately 35 and 4.5 times larger in floorplan than the 2D-3D-S and Matterport3D datasets respectively, however, it currently lacks semantic annotations. Other repositories specialize on different types of semantic information, such as materials (e.g., Materials in Context Database (MINC) [8]), visual/tactile textures (e.g., Describable Textures Dataset (DTD) [12]) and scene categories (e.g., MIT Places [49]).

Automatic and Semi-automatic Semantic Detection Semantic detection is a highly active field (a detailed overview is out of scope for this paper). The main point to stress is that, similar to the repositories, works are focused on a limited semantic information scope. *Object Semantics* range from class recognition to spatial span definition (bounding box/segmentation mask). One of the most recent works is Mask R-CNN [18], which provides object instance segmentation masks in RGB images. Other ones with similar output are Blitz-Net [15] (RGB) and Frustum PointNet [38] (RGB-D).

In addition to detection methods, crowd-sourcing data annotation is a common strategy, especially when building a new repository. Although most approaches focus solely on manual labor, some employ automation to minimize the amount of human interaction with the data and provide faster turnaround. Similar to our approach, Andriluka et al. [4] employ Mask R-CNN trained on the COCO-Stuff dataset to acquire initial object instance segmentation

masks that are subsequently verified and updated by users. Polygon-RNN [9, 2] is another machine-assisted annotation tool which provides contours of objects in images given user-defined bounding boxes. Both remain in the 2D world and focus on object category and segmentation mask.

Others employ lower-level automation to accelerate annotations in 3D. ScanNet [13] proposes a web-interface for manual annotation of 3D mesh models of indoor spaces. It begins with an over-segmentation of the scene using a graph-cut based approach. Users are then prompted to label these segments with the goal of object instance segmentation. [37] has a similar starting point; the resulting over-segments are further grouped into larger regions based on geometry and appearance cues. These regions are edited by users to get object semantic annotations. [41] employs object segmentation masks and labels from 2D annotations to automatically recover the 3D scene geometry.

Despite the incorporation of automation, the above methods largely rely on human interaction to achieve sufficiently accurate results.

3. 3D Scene Graph Structure

The **input** to our method is the typical output of 3D scanners and consists of 3D mesh models, registered RGB panoramas and the corresponding camera parameters, such as the data in Matterport3D [10] or Gibson [44] databases.

The **output** is the 3D Scene Graph of the scanned space, which we formulate as a four-layered graph (see Figure 1). Each layer has a set of *nodes*, each node has a set of *attributes*, and there are *edges* between nodes which represent their relationships. The first layer is the entire *building* and includes the root node for a given mesh model in the graph (e.g., a residential building). The *rooms* of the building compose the second layer of 3D Scene Graph, and each room is represented with a unique node (e.g., a living room). *Objects* within the rooms form the third layer (e.g., a chair or a wall). The final layer introduces *cameras* as part of the graph: each camera location is a node in 3D and a possible observation (e.g., an RGB image) is associated with it.
Attributes: Each building, room, object and camera node in the graph - from now on referred to as element - has a set of attributes. Some examples are the object class, 3D shape, material type, pose information, and more.
Relationships: Connections between elements are established with edges and can span within or across different layers (e.g., object-object, camera-object-room, etc.). A full list of the attributes and relationships is in Table 1.

4. Constructing the 3D Scene Graph

To construct the 3D Scene Graph we need to identify its elements, their attributes, and relationships. Given the number of elements and the scale, annotating the input RGB and
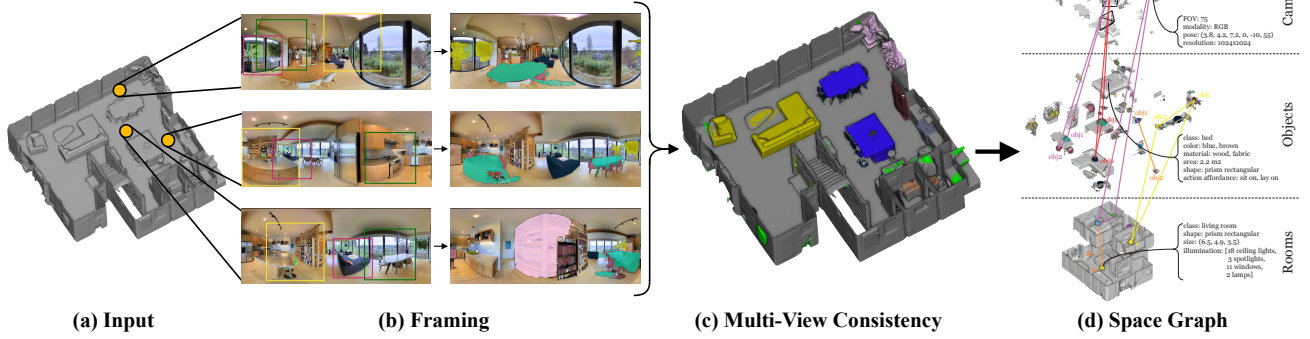
(a) Input	**(b) Framing**	**(c) Multi-View Consistency**	**(d) Space Graph**

Figure 2. **Constructing the 3D Scene Graph.** (a) Input to the method is a 3D mesh model with registered panoramic images. (b) Each panorama is densely sampled for rectilinear images. Mask R-CNN detections on them are aggregated back on the panoramas with a weighted majority voting scheme. (c) Single panorama projections are then aggregated on the 3D mesh. (d) These detections become the nodes of 3D Scene Graph. A subsequent automated step calculates the remaining attributes and relationships.

3D mesh data with object labels and their spatial span is the major labor bottleneck of constructing the 3D Scene Graph. Hence the primary focus of this paper is on addressing this issue by presenting an automatic method that uses existing semantic detectors to bootrstap the annotation pipeline and minimize human labor. An overview of the pipeline is shown in Figure 2. In our experiments (Section 5), we used the best reported performing Mask R-CNN network [18] and got results only for detections with a confidence score of 0.7 or higher. However, since detection results are imperfect, we propose two robustification mechanisms to increase their performance, namely *framing* and *multi-view consistency*, that operate on the 2D and 3D domains respectively.

Table 1. **3D Scene Graph Attributes and Relationships**

Elements	Attributes	Relationships
Object (O)	Action Affordance, Area, Class, ~~Color~~, ID, Location, Material, Occupancy, Shape, Size, Spatial Span, Tactile Texture, Visual Texture, Volume	Amodal Mask *(O,C)*, Parent Space *(O,S)*, Occlusion Relationship *(O,O,C)*, Same Parent Room *(O,O,S)*, Spatial Order *(O,O,C)* Relative Magnitude *(O,O)*
Space (S)	Area, ID, ~~Illumination~~, Location, Occupancy, Scene Category, Shape, Size, Spatial Span, Volume	Spatial Order *(S,S,C)*, Parent Building *(S,B)*, Relative Magnitude *(S,S)*
Building (B)	Area, Building Reference Center, Function, ID, Number of Floors, Shape, Size, Volume	
Camera(C)	Field Of View, ID, Modality, Pose, Resolution	Parent Space (C,S)

Framing on Panoramic Images 2D semantic algorithms operate on rectilinear images and one of the most common errors associated with their output is incorrect detections for partially captured objects at the boundaries of the images. When the same objects are observed from a slightly different viewpoint that places them closer to the center of the image and does not partially capture them, the detection accuracy is improved. Having RGB panoramas as input gives the opportunity to formulate a framing approach that samples rectilinear images from them with the objective to maximize detection accuracy. This approach is summarized in

Figure 3. It utilizes two heuristics: (a) placing the object at the center of the image and (b) having the image properly zoomed-in around it to provide enough context. We begin by densely sampling rectilinear images on the panorama with different yaw (ψ), pitch (θ) and Field of View (FoV) camera parameters, with the goal of having at least one image that satisfies the heuristics for each object in the scene:

$$\psi = [-180°, 180°, 15°], \theta = [-15°, 15°, 15°]$$

$$FoV = [75°, 105°, 15°]$$

This results in a total of 225 images of size 800 by 800 pixels per panorama and serves as the input to Mask-RCNN. To prune out imperfections in the rectilinear detection results, we aggregate them on the panorama using a weighted voting scheme where the weights take into account: the predictions' confidence score and the distance of the detection from the center of the image. In specific, we compute weights per pixel for each class as follows:

$$w_{i,\lambda} = \sum_{j,L_{d_{ij}}=\lambda} \frac{S_{d_{ij}}}{\|C_{d_{ij}} - C_j\|}$$

where $w_{i,\lambda}$ is the weight of panorama pixel i for class λ, $L_{d_{ij}}$ is the class of detection d_{ij} for i in rectilinear frame j, $S_{d_{ij}}$ is the confidence score and $C_{d_{ij}}$ is the center pixel location for the detection, and C_j is the center of j. Given these weights, we compute the highest scoring class per pixel. However, performing the aggregation on individual pixels can result to local inconsistencies, since it disregards information on which pixels could belong to an object instance. Thus, we look at each rectilinear detection and use the highest scoring classes of the contained panorama pixels as a pool of candidates for their final label. We assign the one that is the most prevalent among them. At this stage, the panorama is segmented per class, but not per instance. To
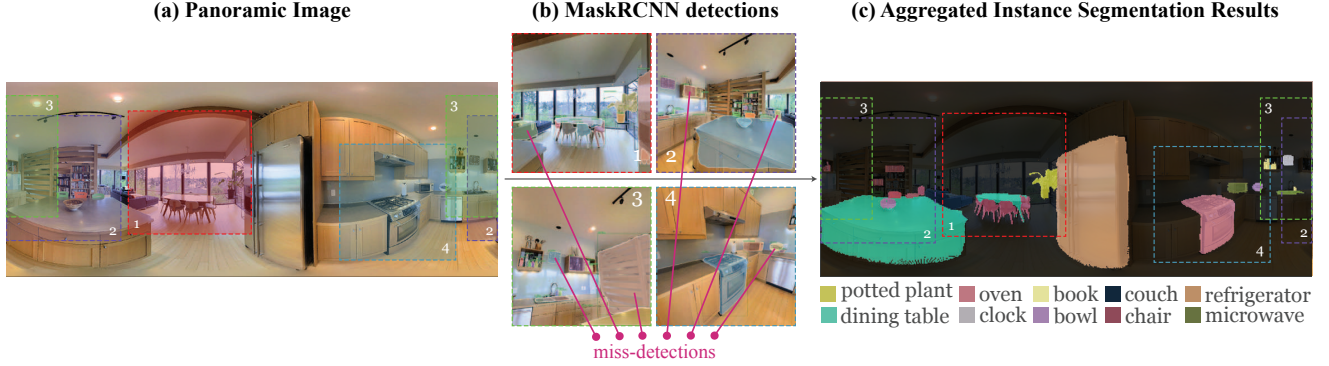
(a) Panoramic Image **(b) MaskRCNN detections** **(c) Aggregated Instance Segmentation Results**

■ potted plant	■ oven	■ book	■ couch	■ refrigerator
■ dining table	■ clock	■ bowl	■ chair	■ microwave

miss-detections

Figure 3. **Framing:** Examples of sampled rectilinear images using the framing robustification mechanism are shown in the dashed colored boxes. Detections (b) on individual frames are not error-free (miss-detections are shown with arrows). The errors are pruned out with weighted majority voting to get the final panorama labels.

address this, we find the per-class connected components; this gives us instance segmentation masks.

Multi-view consistency With the RGB panoramas registered on the 3D mesh, we can annotate it by projecting the 2D pixel labels on the 3D surfaces. However, a mere projection of a single panorama does not yield accurate segmentation, because of imperfect panorama results (Figure 4(b)), as well as common poor reconstruction of certain objects or misalignment between image pixels and mesh surfaces (camera registration errors). This leads to labels "leaking" on neighboring objects (Figure 4(c)). However, the objects in the scene are visible from multiple panoramas, which enables using multi-view consistency to fix such issues. This makes our second robustification mechanism. We begin by projecting all panorama labels on the 3D mesh surfaces. To aggregate the casted votes, we formulate a weighted majority voting scheme based on how close an observation point is to a surface, following the heuristic that the closer the camera to the object, the larger and better visible it is. In specific, we define weights as:

$$w_{i,j} = \frac{\sum_{i,j} \|P_i - F_{c_j}\|}{\|P_i - F_{c_j}\|}$$

where $w_{i,j}$ is the weight of a face F_j with respect to a camera location P_i and F_{c_j} is the 3D coordinates of F_j's center.

Similar to the framing mechanism, voting is performed on the detection level. We look for label consistency across the group of faces F_{obj} that receives votes from the same object instance in a panorama. We first do weighted majority voting on individual faces to determine the pool of label candidates for F_{obj} as it results from casting all panoramas, and then use the one that is most present to assign it to the group. A last step of finding connected components in 3D gives us the final instance segmentation masks. This information can be projected back on the panoramas, hence providing consistent 2D and 3D labels.
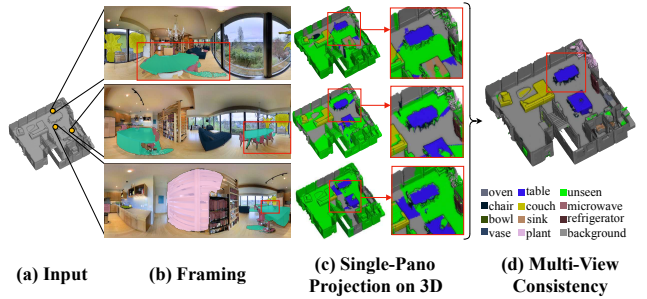

■ oven	■ table	■ unseen	
■ chair	■ couch	■ microwave	
■ bowl	■ sink	■ refrigerator	
■ vase	■ plant	■ background	

(a) Input **(b) Framing** **(c) Single-Pano Projection on 3D** **(d) Multi-View Consistency**

Figure 4. **Multi-view consistency:** Semantic labels from different panoramas are combined on the final mesh via multi-view consistency. Even though the individual projections carry errors from the panorama labels and poor 3D reconstruction/camera registration, observing the object from different viewpoints can fix them.

4.1. User-in-the-loop verification

As a final step, we perform manual verification of the automatically extracted results. We develop web interfaces with which users verify and correct them when necessary. Screenshots and more details on this step are offered in [5]. We crowd-sourced the verification in Amazon Mechanical Turk (AMT). However, we do not view this as a crucial step of the pipeline as the automated results without any verification are sufficiently robust to be of certain practical uses (see Section 5.3 and [5]). The manual verification is performed mostly for evaluation purposes and forming error-free data for certain research use cases.

The pipeline consists of two main steps (all operations are performed on rectilinear images). *Verification and editing:* After projecting the final 3D mesh labels on panoramas, we render rectilinear images that show each found object in the center and to its fullest extent, including 20% surrounding context. We ask users to (a) verify the label of the shown object - if wrong, the image is discarded from the rest of the process; (b) verify the object's segmentation

mask; if the mask does not fulfill the criteria, users (c) add a new segmentation mask. *Addition of missing objects:* The previous step refines our automatic results, but there may still be missing objects. We project the verified masks back on the panorama and decompose it in 5 overlapping rectilinear images (72° of yaw difference per image). This step (a) asks users if any instance of an object category is missing, and if found incomplete, (b) they recursively add masks until all instances of the object category are masked out.

4.2. Computation of attributes and relationships

The described approach gives as output the object elements of the graph. To compute attributes and relationships, we use off-the-shelf learning and analytical methods. For a full description of them and for a video with results see [5].

5. Experiments

We evaluate our automatic pipeline on the Gibson Environment's [44] database.

5.1. Dataset Statistics

The Gibson Environment's database consists of 572 full buildings. It is collected from real indoor spaces and provides for each building the corresponding 3D mesh model, RGB panoramas and camera pose information[1]. We annotate with our automatic pipeline all 2D and 3D modalities, and manually verify this output on Gisbon's tiny split. The semantic categories used come from the COCO dataset [33] for objects, MINC [8] for materials, and DTD [12] for textures. A more detailed analysis of the dataset and insights per attributes and relationships is in [5].

5.2. Evaluation of Automated Pipeline

We evaluate our automated pipeline both on 2D panoramas and 3D mesh models. We follow the COCO evaluation protocol [33] and report the average precision (AP) and recall (AR) for both modalities. We use the best off-the-shelf Mask R-CNN model trained on the COCO dataset. Specifically, we choose Mask R-CNN with Bells & Whistles from Detectron [1]. According to the model notes, it uses a ResNeXt-152 (32x8d) [45] in combination with a Feature Pyramid Network (FPN) [32]. It is pre-trained on ImageNet-5K and fine-tuned on COCO. For more details on implementation and training/testing we refer the reader to Mask R-CNN [18] and Detectron [1].

Baselines: We compare the following approaches in 2D:

- Mask R-CNN [18]: We run Mask R-CNN on 6 rectilinear images sampled on the panorama with no overlap. The detections are projected back on the panorama.

- Mask R-CNN with Framing: The panorama results here are obtained from our first robustification mechanism.
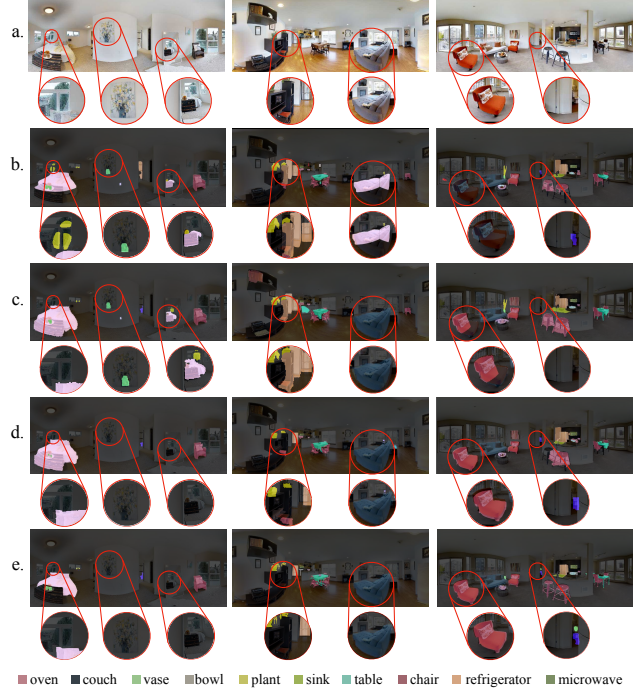

Figure 5. **Detection results on panoramas:** (a) Image, (b) Mask R-CNN [18], (c) Mask R-CNN w/ Framing, (d) Mask R-CNN w/ Framing and Multi-View Consistency (our final results), (e) Ground Truth (best viewed on screen). For larger and additional visualizations see [5].

- Mask R-CNN with Framing and Multi-View Consistency (MVC) - ours: This is our automated method. The panorama results are obtained after applying both robustification mechanisms.

And these in 3D:

- Mask R-CNN [18] and Pano Projection: The panorama results of Mask R-CNN are projected on the 3D mesh surfaces with simple majority voting per face.

- Mask R-CNN with Framing and Pano Projection: The panorama results from our first mechanism follow a similar 2D-to-3D projection and aggregation process.

- Mask R-CNN with Framing and Multi-View Consistency (MVC) - ours: This is our automated method.

As shown in Table 2, each mechanism in our approach contributes an additional boost in the final accuracy. This is also visible in the qualitative results, with each step further removing erroneous detections. For example, in the first column of Figure 5, Mask R-CNN (b) detected the trees outside the windows as *potted plants*, a *vase* on a painting and a *bed* reflection in the mirror. Mask R-CNN with framing (c) was able to remove the tree detections and recuperate a missed *toilet* that is highly occluded. Mask R-CNN with framing and multi-view consistency (d) further removed the

Table 2. **Evaluation of the automated pipeline on 2D panoramas and 3D mesh.** We compute Average Precision (AP) and Average Recall (AR) for both modalities based on COCO evaluation [33].

Method	2D			3D		
	Mask R-CNN [18]	Ours Mask R-CNN w/ Framing	Ours Mask R-CNN w/ Framing + MVC	Mask R-CNN + Pano Projection	Ours Mask R-CNN w/ Framing + Pano Projection	Ours Mask R-CNN w/ Framing + MVC
AP	0.079	0.160	**0.485**	0.222	0.306	**0.409**
AP.50	0.166	0.316	**0.610**	0.445	0.539	**0.665**
AP.75	0.070	0.147	**0.495**	0.191	0.322	**0.421**
AR	0.151	0.256	**0.537**	0.187	0.261	**0.364**

painted vase and bed reflection, achieving results very close to the ground truth. Similar improvements can be seen in the case of 3D (Figure 6). Even though they might not appear as large quantitatively, they are crucial for getting consistent 3D results with most changes relating to consistent local regions and better object boundaries.

Human Labor: We perform a user study to associate detection performance with human labor (hours spent). The results are in Table 3. Note that the hours reported for the fully manual 3D annotation [7] are computed for 12 object classes (versus 62 in ours) and for an *expert* 3D annotator (versus non-skilled labor in ours).

Table 3. **Mean time spent by human annotators per model.** Each step is done by 2 users independently for cross checking.

Method	Ours w/o human (FA)	Ours w/ human (MV)	Human only (FM 2D)	[7] (FM 3D)
AP	0.389	0.97	1	1
Time (h)	0	03:18:02	12:44:10	10:18:06

FA: fully automatic — FM: fully manual — MV: mavual verification

Using different detectors: Until this point we have been using the best performing Mask R-CNN network with a 41.5 reported AP on COCO [18]. We want to further understand the behavior of the two robustification mechanisms when using a less accurate detector. To this end, we perform another set of experiments using BlitzNet [15], a network with faster inference but worse reported performance on the COCO dataset (AP 34.1). We notice that the results for both detectors provide a similar relative increase in AP among the different baselines (Table 4). This suggests that the robustification mechanisms can provide similar value in increasing the performance of standard detectors and correct errors, regardless of initial predictions.

5.3. 2D Scene Graph Prediction

So far we focused on the automated detection results. These will next go through an automated step to generate the final 3D Scene Graph and *compute attributes and relationships*. Results on this can be seen in [5]. We use this output for experiments on 2D scene graph prediction.

There are 3 standard evaluation setups for 2D scene graphs (SG) [35]: (a) SG Detection, (b) SG Classification, and (c) Relationship Classification (for an explanation

chair ■ couch ■ microwave ■ vase ■ plant ■ background ■ oven ■ table ■ unseen ■ bowl ■ sink ■ refrigerator
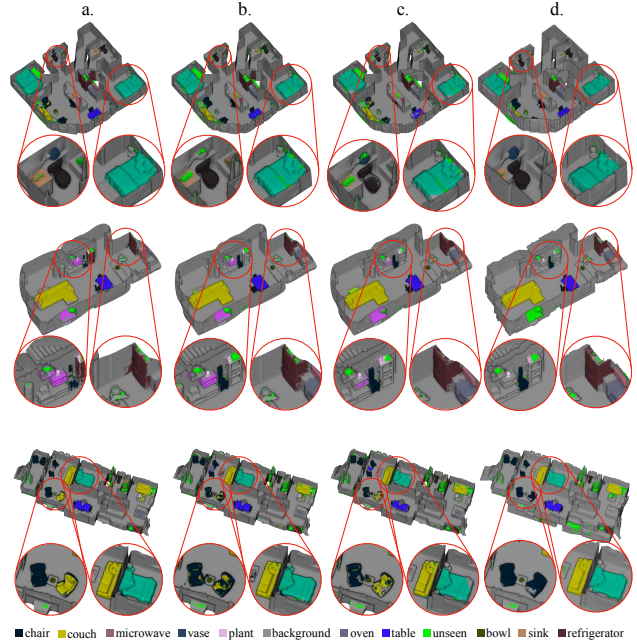
Figure 6. **3D detection results on mesh models:** (a) Mask R-CNN [18] + Pano Projection, (b) Mask R-CNN w/ Framing + Pano Projection, (c) Mask R-CNN w/ Framing and Multi-View Consistency (our final results), (d) Ground Truth (best viewed on screen). For larger and additional visualizations see [5].

see [5]). In contrast to Visual Genome where only sparse and instance-specific relationships exist, our graph is dense, hence some of the evaluations (e.g., relationship detection) are not applicable. We focus on *relationship classification* and provide results on: (a) *spatial order* and (b) *relative volume* classification, as well as on (c) *amodal mask segmentation* as an application of the occlusion relationship.

Spatial Order: Given an RGB rectilinear image and the (visible) segmentation masks of an object pair, we predict if the query object is in front/behind, to the left/right of the other object. We train a ResNet34 using the segmentation masks that were automatically generated by our method, and use the medium Gibson data split. The baseline is Statistically Informed Guess extracted from the training data.

Relative Volume: We follow the same setup and predict

Table 4. **AP performance of using different detectors.**

Method	Detector	Detector w Framing	Detector w/ Framing + MVC	Detector + Pano projection	Detector w Framing + Pano projection	Detector w/ Framing + MVC
Mask R-CNN [18]	0.079	+0.081	**+0.406**	0.222	+0.084	**+0.187**
BlitzNet [15]	0.095	+0.103	**+0.189**	0.076	+0.089	**+0.169**

Figure 7. **Classification results of scene graph relationships.**

whether the query object is smaller or larger in volume than the other object. Figure 7 shows results of predictions for both tasks, whereas quantitative evaluations are in Table 5.

Table 5. **Mean AP for SG Predicate Classification.**

SG Predicate	Baseline	Ours
Spatial Order	0.255	**0.712**
Relative Volume	0.555	**0.820**

Amodal Mask Segmentation: We predict the 2D amodal segmentation of an object partially occluded by others given a camera location. Since our semantic information resides in 3D space, we can infer the full extents of object occlusions without additional annotations and in a fully automatically way, considering the difficulties of data collection in previous works [29, 52, 16]. We train a U-Net [40] agnostic to semantic class, to predict per-pixel segmentation of visible/occluded mask of an object centered on an RGB image (*Amodal Prediction (Ours)*). As baselines, we take an average of amodal masks (a) over the training data (*Avg. Amodal Mask*) and (b) per-semantic class assuming its perfect knowledge at test time (*Avg. Class Specific Amodal Mask*). More information on data generation and experimental setup is in [5]. We report f1-score and intersection-over-union as a per-pixel classification of three semantic classes (empty, occluded, and visible) along with the macro average (Table 6). Although the performance gap may not look significant due to a heavy bias of empty class, our approach consistently shows significant performance boost in predicting occluded area, demonstrating that it successfully learned amodal perception unlike baselines (Figure 8).

6. Conclusion

We discussed the grounding of multi-modal 3D semantic information in a unified structure that establishes relation-

Figure 8. **Sample results of amodal mask segmentation with *Amodal Prediction (Ours)*.** blue: visible, red: occluded.

Table 6. **Amodal mask segmentation quantitative results.**

f1-score	empty	occluded	visible	avg
Avg. Amodal Mask	0.934	0.000	0.505	0.479
Avg. Class Specific Amodal Mask	0.939	0.097	0.599	0.545
Amodal Prediction (Ours)	**0.946**	**0.414**	**0.655**	**0.672**
IoU	empty	occluded	visible	avg
Avg. Amodal Mask	0.877	0.0	0.337	0.405
Avg. Class Specific Amodal Mask	0.886	0.051	0.427	0.455
Amodal Prediction (Ours)	**0.898**	**0.261**	**0.488**	**0.549**

ships between objects, 3D space, and camera. We find that such a setup can provide insights on several existing tasks and allow new ones to emerge in the intersection of semantic information sources. To construct the 3D Scene Graph, we presented a mainly automatic approach that increases the robustness of current learning systems with *framing* and *multi-view consistency*. We demonstrated this on the Gibson dataset, which 3D Scene Graph results are publicly available. We plan to extend the object categories to include more objects commonly present in indoor scenes, since current annotations tend to be sparse in places.

References

[1] Detectron model zoo. https://github.com/facebookresearch/Detectron/blob/master/MODEL_ZOO.md. Accessed: 2019-08-12.

[2] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with

polygon-rnn++. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 859–868, 2018.

[3] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016.

[4] Mykhaylo Andriluka, Jasper RR Uijlings, and Vittorio Ferrari. Fluid annotation: a human-machine collaboration interface for full image annotation. *arXiv preprint arXiv:1806.07527*, 2018.

[5] Iro Armeni, Jerry He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. Supplementary material for: 3d scene graph: A structure for unified semantics, 3d space, and camera. `http://3dscenegraph.stanford.edu/images/supp_mat.pdf`. Accessed: 2019-08-16.

[6] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017.

[7] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543, 2016.

[8] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the materials in context database. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3479–3487, 2015.

[9] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a polygon-rnn. In *CVPR*, volume 1, page 2, 2017.

[10] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

[11] Wongun Choi, Yu-Wei Chao, Caroline Pantofaru, and Silvio Savarese. Understanding indoor scenes using 3d geometric phrases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 33–40, 2013.

[12] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

[13] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas A Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, volume 2, page 10, 2017.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[15] Nikita Dvornik, Konstantin Shmelkov, Julien Mairal, and Cordelia Schmid. Blitznet: A real-time deep network for scene understanding. In *ICCV 2017-International Conference on Computer Vision*, page 11, 2017.

[16] Kiana Ehsani, Roozbeh Mottaghi, and Ali Farhadi. Segan: Segmenting and generating the invisible. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6144–6153, 2018.

[17] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[19] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. Holistic 3d scene parsing and reconstruction from a single rgb image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 187–203, 2018.

[20] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317, 2016.

[21] Chenfanfu Jiang, Siyuan Qi, Yixin Zhu, Siyuan Huang, Jenny Lin, Lap-Fai Yu, Demetri Terzopoulos, and Song-Chun Zhu. Configurable 3d scene synthesis and 2d image rendering with per-pixel ground truth using stochastic grammars. *International Journal of Computer Vision*, 126(9):920–941, 2018.

[22] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017.

[23] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4565–4574, 2016.

[24] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.

[25] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[26] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 3337–3345. IEEE, 2017.

[27] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[28] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[29] Ke Li and Jitendra Malik. Amodal instance segmentation. In *European Conference on Computer Vision*, pages 677–693. Springer, 2016.

[30] Yikang Li, Wanli Ouyang, Bolei Zhou, Kun Wang, and Xiaogang Wang. Scene graph generation from objects, phrases and region captions. In *ICCV*, 2017.

[31] Xiaodan Liang, Lisa Lee, and Eric P Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4408–4417. IEEE, 2017.

[32] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[34] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European Conference on Computer Vision*, pages 852–869. Springer, 2016.

[35] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European Conference on Computer Vision*, pages 852–869. Springer, 2016.

[36] Tomasz Malisiewicz and Alyosha Efros. Beyond categories: The visual memex model for reasoning about object relationships. In *Advances in neural information processing systems*, pages 1222–1230, 2009.

[37] Duc Thanh Nguyen, Binh-Son Hua, Lap-Fai Yu, and Sai-Kit Yeung. A robust 3d-2d interactive tool for scene segmentation and annotation. *IEEE transactions on visualization and computer graphics*, 24(12):3005–3018, 2018.

[38] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. *arXiv preprint arXiv:1711.08488*, 2017.

[39] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. *arXiv preprint arXiv:1808.07962*, 2018.

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[41] Bryan C Russell and Antonio Torralba. Building a database of 3d scenes from user annotations. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2711–2718. IEEE, 2009.

[42] Gabriel Sepulveda, Juan Carlos Niebles, and Alvaro Soto. A deep learning based behavioral approach to indoor autonomous navigation. *arXiv preprint arXiv:1803.04119*, 2018.

[43] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *3D Vision (3DV), 2017 International Conference on*, pages 537–547. IEEE, 2017.

[44] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9068–9079, 2018.

[45] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[46] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2017.

[47] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. Visual translation embedding network for visual relation detection. In *CVPR*, volume 1, page 5, 2017.

[48] Yibiao Zhao and Song-Chun Zhu. Scene parsing by integrating function, geometry and appearance models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3119–3126, 2013.

[49] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.

[50] Yuke Zhu, Alireza Fathi, and Li Fei-Fei. Reasoning about object affordances in a knowledge base representation. In *European conference on computer vision*, pages 408–424. Springer, 2014.

[51] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4995–5004, 2016.

[52] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1464–1472, 2017.