Unknown Object Segmentation from Stereo Images

Maximilian Durner*[1], Wout Boerdijk*[1], Martin Sundermeyer[1],
Werner Friedl[1], Zoltán-Csaba Márton[3], Rudolph Triebel[1,2]

Abstract— **Although instance-aware perception is a key prerequisite for many autonomous robotic applications, most of the methods only partially solve the problem by focusing solely on known object categories. However, for robots interacting in dynamic and cluttered environments, this is not realistic and severely limits the range of potential applications. Therefore, we propose a novel object instance segmentation approach that does not require any semantic or geometric information of the objects beforehand. In contrast to existing works, we do not explicitly use depth data as input, but rely on the insight that slight viewpoint changes, which for example are provided by stereo image pairs, are often sufficient to determine object boundaries and thus to segment objects. Focusing on the versatility of stereo sensors, we employ a transformer-based architecture that maps directly from the pair of input images to the object instances. This has the major advantage that instead of a noisy, and potentially incomplete depth map as an input, on which the segmentation is computed, we use the original image pair to infer the object instances and a dense depth map. In experiments in several different application domains, we show that our *Instance Stereo Transformer* (INSTR) algorithm outperforms current state-of-the-art methods that are based on depth maps. Training code and pretrained models are available at `https://github.com/DLR-RM/instr`.**

I. INTRODUCTION

Robots interacting in real-world environments are often faced with a large variety of object instances. Acquiring the information necessary for successful interaction with these objects is partly addressed by the field of object instance recognition, where large advances were made in terms of accuracy and robustness. Nevertheless, the majority of existing methods require prior knowledge in terms of annotated data or 3D models for each considered object class or instance.

This work presents a novel stereo-based approach for Unknown Object Instance Segmentation (UOIS) to address the mentioned issue in robotic vision. Starting from the transformer encoder-decoder structure proposed in DETR [1] we modify the cross-attention mechanism in the decoder to directly predict instance segmentation masks without the intermediate detection step. Our method, which is purely trained on synthetic images, is able to predict unknown objects on generic horizontal surfaces in an end-to-end manner without any post-processing due to the set-prediction attribute of the applied transformer structure.

As shown in [3], [4] depth information is a crucial modality for robust UOIS. However, affordable depth sensors still cannot deal with untextured surfaces (stereo sensors), metallic, transparent or black materials and bright light (active sensors), which results in noisy and incomplete depth maps. Modeling imperfect depth data is difficult, sensor specific and thus often tackled by augmentations [3], [4]. But are random augmentations on synthetic depth the best way for the network to decide when to rely on depth and when on RGB information at test time? Additionally, the fusion of RGB and corrupt depth data early in the network is non-trivial [5]. We hypothesize that a network that simultaneously learns disparity from stereo can build up a richer internal representation to determine where depth cues are useful and trustworthy and where it is better to follow textural information.

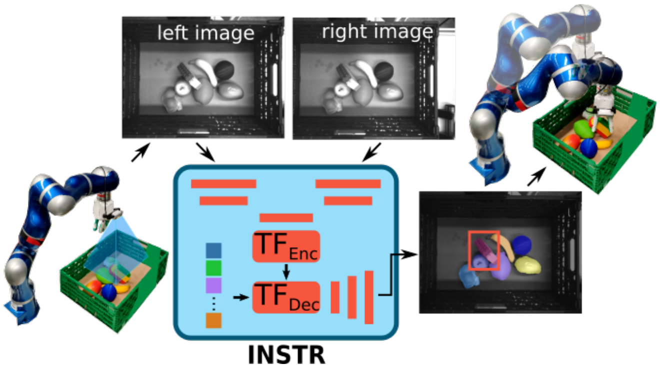


Fig. 1: Given a stereo image pair, INSTR segments unknown object instances on generic horizontal surfaces. The obtained pixel-wise object masks can be used for grasping, here exemplary with the clash hand [2]. The as auxiliary task predicted depth can further aid the robot in interacting with its environment.

While depth data is beneficial in robotic applications, human-level segmentation of novel objects should be possible to learn from stereo image pairs alone, since we lack the capability of high-precision, active depth perception. We calibrate our depth perception by reaching and walking [6] well after we learn to recognize different objects and continuously correct the errors we make during interaction with them [7].

Concretely, we contribute the following:

- We propose an end-to-end stereo-based approach that jointly learns disparity and unknown object instance segmentation from physically-based RGB stereo renderings. Experiments show that this is a promising approach to break the reliance on high quality depth data for robust unknown object segmentation.
- We further introduce a sub-pixel sampling mechanism for our correlation layers, which enables a dynamic adaptation to other stereo sensor settings (e.g. changing

*Equal contribution

[1]Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Wessling, Germany `<first>.<second>@dlr.de`

[2]Department of Computer Science, Technical University of Munich (TUM), 85748 Garching, Germany

[3]Agile Robots AG, 81477 Munich, Germany (work performed at DLR)

baseline) the network was not trained on.

- We address the absence of an instance-centric stereo dataset and introduce *Stereo Instances On Surfaces* (STIOS), a binocular dataset consisting of 192 scenes from two different stereo sensors on various surfaces with manually labelled, pixel-wise instance masks.
- We adapt the DETR [1] transformer architecture for the task of instance segmentation. In concrete terms, by adapting the cross-attention mechanism as well as the segmentation loss, our transformer outputs 2D queries for direct upsampling. As a result we obtain a post-processing-free class-agnostic instance segmentation pipeline running at 18 frames per second.

II. Related Work

Set-Prediction: Object-centric vision tasks such as (class-agnostic) detection or instance segmentation are usually tackled with CNN-based architectures. However, the arbitrary number of instances in a scene, their permutation invariance as well as the responsibility problem [8] are not directly solved by CNNs [1]. Instead, for these set-prediction problems, convolutional architectures usually generate a large number of proposals with hand-designed anchors that are subsequently refined. Even anchor-free methods [9], [10] then require expensive post-processing steps such as Non-Maximum Suppression [11], watershed algorithm [12], hough-voting [13], or clustering [14], [15] to filter the raw network outputs. Besides introducing additional hyperparameters, these post-processing methods take a large portion of the inference time when considering lightweight architectures that are crucial for robotic systems.

Direct, i.e post-processing-free, set-prediction with CNNs requires adaptations, e.g. an autoregressive or recurrent structure to process scenes instance-by-instance [16], [17]. Greff *et al.* [18] iteratively infer a set of latent representations, each representing an object, by variational inference. In a similar manner, multiple encoder-decoder steps are applied in [19], [20] to refine object-centric representations. Zhang *et al.* [8] present a general set-prediction approach which optimizes the mean squared error between a latent representation of the input (image) and a set of feature vectors. An iterative attention module is presented in [21] which groups task-specific input features to a set of output vectors. While these experiments have indicated the potential of direct set-predictions, they haven't proven their applicability to complex real world applications, yet.

Recently, transformer networks [22], originally used for NLP tasks, gained a lot of attention in computer vision. While [23], [24] apply transformers on sequences of image patches for classification, the DETR models [1], [25] directly output a set of bounding box predictions in parallel by cross-attending object queries to the global image context. An earlier work by Liang *et al.* [26] exploits self-attention across polygon vertices to improve the prediction of the offsets. Based on the initial DETR model [1] several works have been published for instance segmentation [27], [28]. These works reuse the original structure and predict the instance masks based on the bounding box features which again resembles an indirect approach. The transformer network presented by Xie *et al.* [29] addresses transparent object segmentation based on RGB. Their object queries encode class-related features which is not applicable to UOIS.

Unknown Object Instance Segmentation: Early works in UOIS mainly build on low-level image features based on boundaries, connectivity or symmetry [30], [31] to segment unknown instances. Richtsfeld *et al.* [32] proposed to estimate surface patches based on a mixture of planes and NURBS on which a graph-cut algorithm is applied. However, such features are often insufficient to model what constitutes an object in more complex settings [33]. To let robots manipulate completely unknown instances we need to learn a concept of "objectness" defined as a geometrically and often semantically connected entity. One way to extract unknown objects from a scene is by predicting their independent motion masks [34]. While most objects are naturally static, robotic manipulators can induce the necessary motion so that grasped objects can be segmented from arbitrary viewpoints [35]. In [36] a class-agnostic segmentation mask together with an object likelihood score is predicted per RGB image patch of the MS COCO dataset [37]. However, color information from a few specific categories does not suffice to learn the "objectness" relevant in robotics contexts. For this purpose, more diverse training data can be generated synthetically by combining large 3D model databases like ShapeNet [38] with procedural data generation methods [39]. A straightforward approach is to train existing instance segmentation methods (e.g. Mask R-CNN [11]) with only a single foreground object category [3], [40]. Xie *et al.* [13] outperform this baseline by predicting 2D unit vectors pointing towards object centers from synthetic point clouds and then refining the predictions with another network using color information. Similar to [14], Xiang *et al.* [4] cluster pixel-wise feature representation predicted by a single convolutional network jointly trained on synthetic RGB and depth data. Instead we jointly learn depth and unknown instance segmentation from photo-realistic stereo RGB renderings and directly predict instance masks with a transformer-based architecture.

Stereo Segmentation: Recent networks are capable of sub-pixel accurate disparity estimation from stereo images [41]. Several works [42]–[44] already show the mutual benefit of jointly learning disparity and semantic segmentation. In this work we investigate whether class-agnostic instance segmentation also benefits from jointly learning of disparity.

III. Method

Our proposed method takes as input a pair of stereo-images and implicitly fuses the disparity and RGB cues to avoid the necessity of high-quality depth data. As Fig. 2 shows, the inputs are forwarded through a feature extraction encoder to obtain a correlated feature representation of both images (Sec. III-A). Next, the features are processed by an instance-aware *Transformer Encoder* TF_{Enc} followed by a *Transformer Decoder* TF_{Dec} separating the instances (Sec. III-B). In contrast to the original TF_{Dec}, ours outputs 2D feature maps which can directly be upsampled. Moreover, another decoder is added to predict auxiliary disparities.

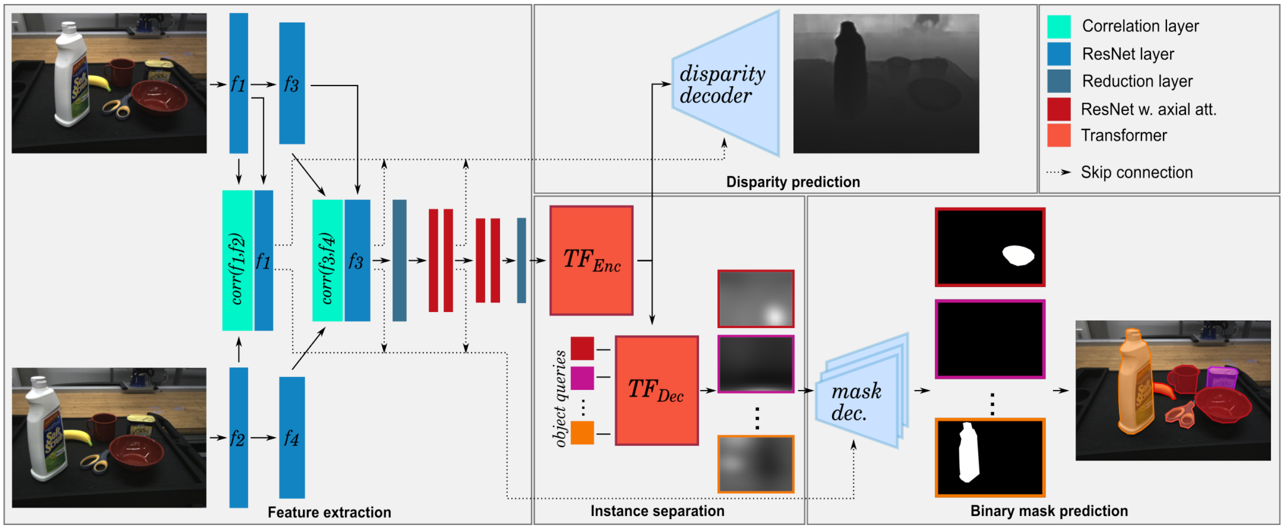


Fig. 2: INSTR consists of an encoder-decoder structure with two local correlation layers in the feature extractor (left), transformer encoder and decoder layers (middle) and two separate decoders for disparity and instance predictions (right).

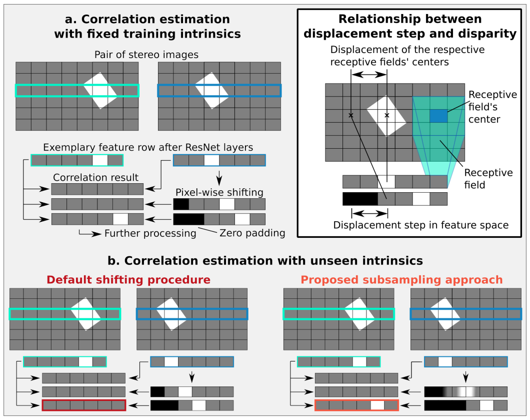

a. Correlation estimation with fixed training intrinsics
Pair of stereo images
Exemplary feature row after ResNet layers
Correlation result
Pixel-wise shifting
Further processing
Zero padding
Relationship between displacement step and disparity
Displacement of the respective receptive fields' centers
Receptive field's center
Receptive field
Displacement step in feature space
b. Correlation estimation with unseen intrinsics
Default shifting procedure
Proposed subsampling approach

Fig. 3: Overview of our local correlation layer (a); our proposed subsampling generalizes to novel camera intrinsics (b); relationship between displacement step in feature space and disparity in image plane (top right).

A. *Feature Extraction Encoder*

A stereo pair is forwarded through the first two layers of a ResNet-50 [45] backbone with shared weights. After each, a correlation layer [46], [47] restricted to a local horizontal region to capture stereoscopic information is applied. Mathematically, given the feature maps $\mathbf{f}_a, \mathbf{f}_b \in \mathbb{R}^{cxhxw}$ of a stereo pair with c, h, w corresponding to number of channels, height, and width, we define the *local horizontal correlation* at a specific spatial position $\mathbf{x}_a$ of $\mathbf{f}_a$ and $\mathbf{x}_b$ of $\mathbf{f}_b$ as

$$corr(\mathbf{x}_a, \mathbf{x}_b) = \sum_{i=0}^{d_{max}} \left\langle \mathbf{f}_a(\mathbf{x}_a), \mathbf{f}_b(\mathbf{x}_b + \begin{pmatrix} i * s \\ 0 \end{pmatrix}) \right\rangle \;, \tag{1}$$

with d_{max} being the maximum shift of $\mathbf{f}_b$ in positive horizontal direction and $s = 1$ being the displacement size. This displacement in a downsampled feature map can approximately be related to a certain disparity value given the centers of the respective receptive fields. Note that to have the same width as $\mathbf{f}_a$, $\mathbf{f}_b$ is zero padded on the left. The outcome is a correlation tensor $C \in \mathbb{R}^{c_c xhxw}$ where c_c depicts the total number of displacement steps - a fixed value, since subsequent layers employ convolutional operations. Consequently, the horizontal focal length f_x and baseline b_x are fixed during training. To dissolve this limitation and enable variable sensor intrinsics during inference, we allow continuous values for s and obtain the corresponding $\mathbf{f}_b$ with bilinear grid sampling. Based on the relation with the receptive field, d_{max} can be computed by:

$$d_{\text{max}} \approx \frac{f_c * b_c}{z_{\text{min}} * output\_stride} \;, \tag{2}$$

where z_{min} is the minimum camera-to-object distance and $output\_stride$ denotes the related downsampling ratio. Given (2) we then can generalize to arbitrary intrinsic parameters f_c' and b_c' by calculating the respective d'_{max}, and equally well compensate for a new z'_{min}. The corresponding new step size s' can be derived via the relation:

$$c_c = \frac{d_{\text{max}}}{s} = \frac{d'_{\text{max}}}{s'} \;. \tag{3}$$

The intuition here is that given fixed intrinsics during training the network expects specific correlation results at specific channel positions of c_c. Whenever f_c and/or b_c differ, the respective correlative features are at different positions and result in unexpected and confusing information to the network. Changing d_{max} and s respectively counteracts this behavior and maintains the features at their "correct" channel positions. This adaptation of d_{max} and s can be done without any retraining or fine-tuning since the correlation layer does not involve learnable parameters. Furthermore, the sub-pixel-based sampling allows us to process arbitrary input sizes. Note that d_{max} is set such that the centers of the respective receptive fields cover more than the maximal (or any desired amount of) disparity, or a z_{min} (see Fig. 3).

Since the correlation layer is computationally expensive, the number of channels c of f_a and f_b are reduced beforehand. The correlation output of both layers is concatenated with the respective f_a and passed as a skip connection to the decoder. For the second output the concatenated map is forwarded to subsequent encoder layers such that the transformer obtains correlation information. We employ a reduction layer to fit the subsequent convolutional layer.

For segmentation tasks the stride of the following ResNet layers 3 and 4 is usually replaced with dilation to result in an *output_stride* of 8 instead of 32, which allows denser feature responses to be extracted [48]. While this does not increase the number of learnable parameters in the encoder, it results in an exponential increase in the number of samples for the transformer (300→4,800 for an image of 640x480), which would require more computational resources. To capture meaningful features with *output_stride* $= 32$ we experiment with replacing ResNet layers 3 and 4 with their axial attention counterparts, and refer the reader to [49] for further details. Finally, we channel-wise reduce the features (2048→256) and pass them to the transformer.

B. Transformer Encoder-Decoder

The proposed approach builds up on the DETR architecture with an adapted TF_{Dec}. Based on a transformer encoder-decoder structure, the main component is the attention mechanism, which is briefly described. For further details we refer the reader to [1].

Given a *query sequence* $X_q \in \mathbb{R}^{t\text{x}N_q}$ and a *key-value sequence* $X_{kv} \in \mathbb{R}^{t\text{x}N_{kv}}$, the embeddings query Q, key K and value V can be defined as linear projection: $Q = W_q(X_q + P_q); K = W_k(X_{kv} + P_k); V = W_v X_{kv}$, where $W_k, W_q \in \mathbb{R}^{d_h\text{x}d}$ and $W_v \in \mathbb{R}^{d_{\text{out}}\text{x}d_{\text{in}}}$ are learned weights (in our case $d_{\text{out}} = d_{\text{in}}$; in the following both variables are denoted as d). The terms $P_q \in \mathbb{R}^{d\text{x}N_q}$ and $P_k \in \mathbb{R}^{d\text{x}N_{kv}}$, either learned or fixed, represent positional encodings to maintain spatial information, which is crucial for spatial structures or shapes. In the next step an attention map A is computed by the inner product between the query Q and key K:

$$A = \underset{N_{kv}}{\mathit{softmax}}(Q^T K) \ , \tag{4}$$

where the softmax operation is applied along the N_{kv} dimension. The final output of one attention layer $\texttt{att}$ is then computed by the attention-weighted sum over V. Using the Einstein summation convention (*einsum*) this can be expressed as:

$$\texttt{att}_{N_q d} = A_{N_q N_{kv}} V_{N_{kv} d} \ . \tag{5}$$

For the attention mechanism applied in a TF_{Enc} layer, called *self-attention*, all three embeddings are linear projections of the input map ($X_q = X_{kv}$). On the other hand, the so-called *cross-attention* mechanism $\texttt{c-att}$, applied in a TF_{Dec} layer, attends the resulting $\texttt{att}$ as X_{kv} with Q consisting of a set of N_q embeddings of length t. The set-elements are learnt positional encodings, also called *object queries*. In the TF_{Dec} every instance is represented by one object query based on the instance aware feature sequence generated by TF_{Enc}. Applying (5) for cross-attention, as done originally, results in N_q object queries each with dimension d. While this works for detection, the resulting object queries cannot directly be used in a segmentation decoder as they only represent point-wise features. To circumvent this, [1] attend and concatenate these queries with the image tensor for the task of panoptic segmentation. Similarly, [29] upstream the attention weights of the last cross-attention. Instead, we rewrite the (5) to directly obtain an expanded attention-weighted feature map for each query:

$$\texttt{c-att-exp}_{N_q N_{kv} d} = A_{N_q N_{kv}} V_{N_{kv} d} \ . \tag{6}$$

Adding up along the N_{kv} dimension results again in $\texttt{att}$ (5), used to forward information to the next TF_{Dec} layer, while for the auxiliary loss as well as for upsampling, $\texttt{c-att-exp}$ is considered. Note, the number of computations of the expanded attention is the same as for the standard attention mechanism. The final TF_{Dec} outcome is a set of 2-D object queries $\hat{y} = \{\hat{y}_1, ..., \hat{y}_{N_q}\}$ with $|\hat{y}| = N_q$, where $\hat{y}_i$ is either a binary mask representing an object or a zero mask. As shown in Fig. 2 the object queries are then upsampled by independent *mask decoders* with shared weights.

C. Permutation-Invariant Instance Segmentation Loss

Besides the permutation-invariance of TF_{Dec}, the loss design is also crucial to enhance variety within the embedding set. Given the final output-set $\hat{y}$ a bipartite matching against the ground truth masks $y = \{y_1, ..., y_L\}$ has to be applied. If $|\hat{y}| > |y|$ the difference is padded with zero ground truth masks. In contrast to [1], we directly apply the matching on the segmentation masks. Given the dice loss $\mathcal{L}_{\text{dice}} = 1 - \frac{2y\hat{y}+1}{y+\hat{y}+1}$ as cost function, the optimal assignment is computed by the Hungarian method [50]. Based on the outcome, the actual loss is calculated. In this context a challenging task is to deal with the evaluation of zero masks, which is not considered by the original dice loss. One solution is to treat the background as an object by inverting the maps to $(1 - \hat{y})$ and $(1 - y)$. However, the dice loss is favourable to big masks as a few incorrectly predicted pixels do not lead to a significant change of the loss value. This prevents the network of predicting complete zero masks since almost perfect predictions already achieve decent loss values. To overcome this issue, an exponential logarithmic dice loss [51] to focus more on the least and most correct predictions is used. For a query output $\hat{y}_i$ and its matched ground truth mask $\hat{y}_i$ it can be written as:

$$\mathcal{L}^i_{\text{segm}} = \begin{cases} \mathcal{L}_{\text{dice}}(y^*_l, \hat{y}_i) & \text{if} \sum y^*_l \neq 0 \\ -ln(\mathcal{L}_{\text{dice}}(y^*_l, \hat{y}_i))^\gamma & \text{if} \sum y^*_l = 0 \end{cases} \tag{7}$$

D. Auxiliary Disparity Prediction

In order to provide the network a guidance for an efficient exploitation of stereo cues, an auxiliary decoder is employed. As shown in Fig. 2 the feature map generated by TF_{Enc} and intermediate features with correlation results as skip connections are used to predict the disparity map. For this auxiliary task the *Huber Loss* L_{hub} is employed and the total network loss can be written as

$$\mathcal{L} = \alpha \mathcal{L}_{\text{hub}} + \beta \sum_{j}^{M_{\text{dec}}} \sum_{i}^{N_q} \mathcal{L}^{ij}_{\text{segm}} \ , \tag{8}$$

where α and β are weighting factors, M_{dec} is the number of layers in TF_{Dec} and $\mathcal{L}^{ij}_{\text{segm}}$ indicates the segmentation loss of the ith query in the jth layer.

IV. Stereo Instances on Surfaces Dataset

There exist several datasets of objects on table-top surfaces, and many provide segmentation [32], [52], bounding box [53], [54] or point cloud [55] annotations. Still, to the best of our knowledge none include both stereo images and pixelwise object instance annotations. To address this issue, we present STIOS which consists of stereo images of objects on top of eight tabletop-like surfaces which are situated in various environments. We employ two stereo sensors, a *rc_visard 65 color* and a *Zed* camera, which both capture depth information from stereo and, in the case of Zed, normal directions and point cloud data. For each surface we manually select four camera positions to capture the scene at different distance ranges and elevation levels. Every image depicts a configuration of four to six randomly sampled objects of a subset of the YCB Video dataset [56] [1], where object sampling considers the total number of occurrences per object which should be similar across all items.

Regarding object placement we follow the idea of previous work (e.g. [32], [52]) and differentiate between *simple* (no physical contact between objects) and *difficult* (physically touching or stacked) scenes. Note that objects might appear occluded in the image plane irrespective of the setting. For one camera pose we record three simple and three difficult configurations with both sensors, totaling 6 images per camera pose and 24 images per surface area. All configurations considered, the dataset consists of 192 stereo images for each sensor. To improve the stereo matching for the rc_visard, we additionally project a pattern on the scene. Every left image is manually annotated with corresponding ground truth object instance masks. We believe that this dataset can serve as a reasonable baseline for stereo-aided object instance segmentation of table-top-like scenes.

V. Experiments

A. Simulating Objects on a Table

Due to the lack of a suitable dataset, synthetic training data is generated using BlenderProc [39]. In detail, we select table-top surfaces inside rooms of the SunCG dataset [57], where five to twelve random instances of the ShapeNet dataset [38] are placed in a physical simulation. For each table, ten camera poses within the upper hemisphere of the table's center are sampled with varying distances.[2]

B. Implementation Details

Our training data consists of 40,000 images (90/10 train/val. split) with an input size of 640x480 pixels. We solely train synthetic data (Sec. V-A), and the best model is selected in terms of highest mIoU score on the synthetic validation set. The test data is the realworld STIOS dataset (Sec. IV), which does not include any object from the training or validation data. Regarding training parameters, Tab. I shows the settings for both correlation layers. All experiments are conducted with $N_q = 15$ (mainly due to memory limits), which in general should be larger than the total number of objects in a scene. The exponential factor γ of (7) is set to 0.2, the loss (8) is weighted with $\alpha = \beta = 1$ and optimized by AdamW [58] with a weight decay of 1e-2. With a learning rate of 1e-4 for all trainable parameters and batch size of 2, training for 40 epochs roughly takes 4 days. An inference forward pass on an Nvidia RTX 2080 takes around $55ms$, thus our algorithm can operate at roughly 18fps.

TABLE I: Parameter settings for both correlation layers. The parameter d_{max} corresponds to a minimum camera-to-object distance of $12cm$ for rc_visard. r depicts the downsampling ratio which reduces c of $f_{\{a,b\}}$, and $\hat{D}_{\text{max}}$ the approximate maximum disparity (see Sec. III-A). All values denote pixels.

Layer	h/w	d_{max}	r	c_c	recept. field	$\hat{D}_{\text{max}}$
Corr1	120/160	64	8	64	35x35	260
Corr2	60/80	32	8	32	91x91	264

TABLE II: Mean IoU [%] on object instance masks across all scenes of STIOS. Values in **bold** denote the best results.

Method	rc_visard		rc_visard + pattern		Zed	
	mIoU	F1	mIoU	F1	mIoU	F1
Xie *et al.* [13]	29.25	39.57	44.21	55.21	17.86	25.00
Xiang *et al.* [4]	32.76	39.80	53.74	64.22	15.37	19.72
RGB only	53.63	67.05	*n/a*	*n/a*	49.75	62.69
Depth only	15.16	20.49	25.20	33.21	06.23	08.82
INSTR	**74.93**	**84.50**	*n/a*	*n/a*	**74.06**	**83.80**
Mask-RCNN (YCB-V)	76.43	85.93	*n/a*	*n/a*	68.46	79.22

C. Comparison to the State-of-the-art Methods

Tab. II shows the comparison between our method and various recent baselines on STIOS. Evaluation is done by matching predicted objects with all ground truth instances of an image, and the set with less items is padded with empty masks. As metrics we calculate the Intersection over Union (IoU) and F1 score on the matched pairs, and average across all object instances. Xiang *et al.* [4] clusters object instances in deep feature space to circumvent set prediction, and [13] merges a foreground mask with object center vectors, which is then refined by RGB data. As an upper performance bound we train a Mask-RCNN directly on photorealistically rendered 3D models of the YCB-V dataset used in the BOP Challenge [59]. Exemplary qualitative results on STIOS are depicted in Fig. 4, and Fig. 6 shows the performance of INSTR in the wild.

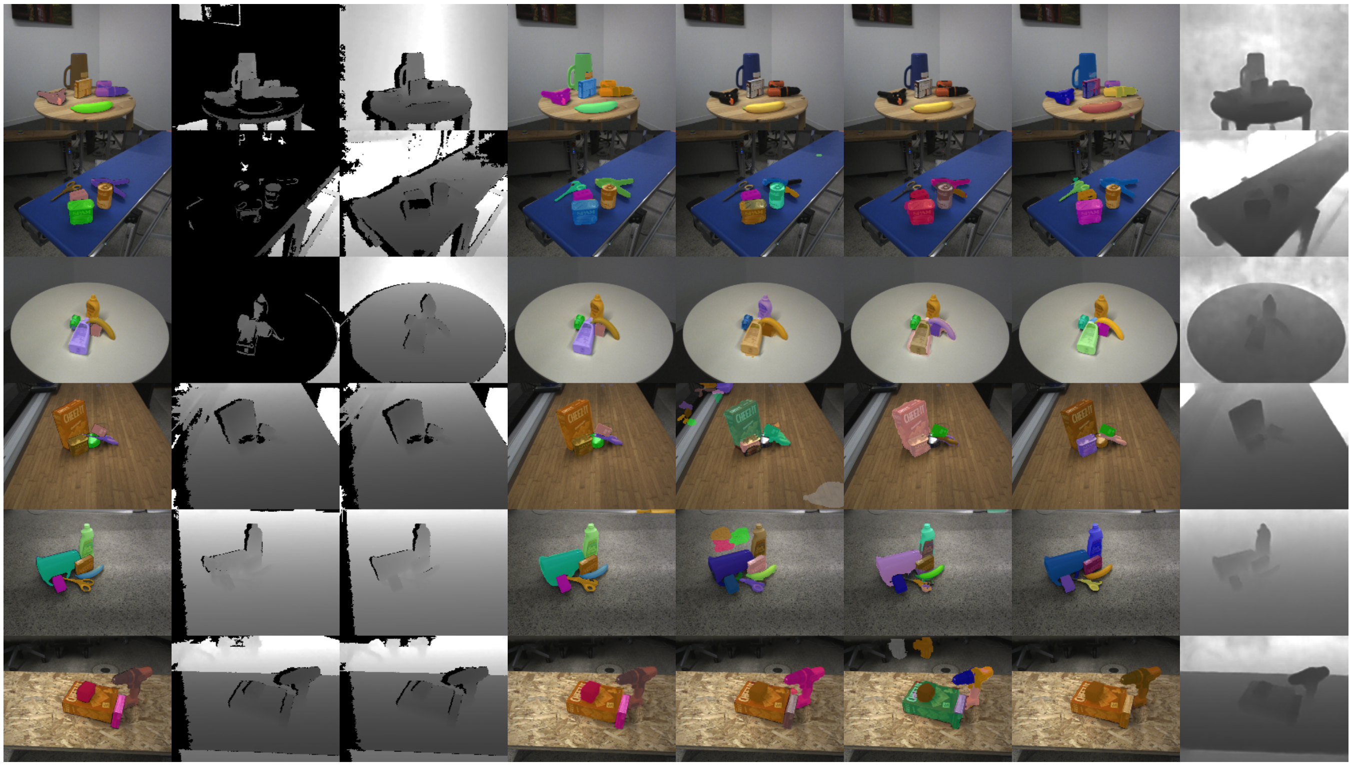
On STIOS the proposed method outperforms all baselines and achieves metrics en par with the Mask-RCNN that has already seen the objects. While the projected pattern increases results for depth-based methods, the obtained data is still insufficient for reliable predictions, possibly because of the amount of noise and incomplete data (total black areas of Fig. 4 b and c).

D. Ablation Studies

1) Architectural Ablation: We continue by exploring the influence of various design choices to our network in Tab. III.

[1] The following objects were used: *003_cracker_box*, *005_tomato_soup_can*, *006_mustard_bottle*, *007_tuna_fish_can*, *008_pudding_box*, *010_potted_meat_can*, *011_banana*, *019_pitcher_base*, *021_bleach_cleanser*, *024_bowl*, *025_mug*, *035_power_drill*, *037_scissors*, *052_extra_large_clamp*, *061_foam_brick*.

[2] For reproduction please see `https://github.com/DLR-RM/instr/blenderproc`



(a) Left RGB+GT (b) Depth w/o pat. (c) Depth w/ pat. (d) Mask-RCNN (e) Xie *et al.* [13] (f) Xiang *et al.* [4] (g) Ours (h) Our depth

Fig. 4: Qualitative results on STIOS (best viewed magnified and in color). Depth-based approaches trained on simulated data (e, f) struggle with real world stereo depth, especially if they contain fragmentary data (b, c). This potentially results in segmentation of background clutter (e: fourth / fifth row, f: last row), undetected objects (second row) or completely empty predictions (first row). INSTR (g) inherently utilizes cues from slight view point changes from a stereo pair and produces depth as auxiliary task (h). The last row denotes a particularly bad case where INSTR fails to separate the objects. The predictions of (e) and (f) are based on depth with projected pattern. The Mask-RCNN (d) is directly trained on 3D models of the YCB-V dataset and presents an upper baseline. Colors are assigned randomly.

Note that for all results in this section we only train up to 15 epochs and stop. Both the axial attention blocks as well as employing an auxiliary loss on upsampled intermediate TF_{Dec} outputs guides in learning meaningful representations; the latter being in coherence to previous work (e.g. [1]). We furthermore compare our proposed query processing method `c-att-exp` (6) with upsampling attention weight maps (`att`, [29]), as well as concatenating these with backbone features (`c-att-cat-bb`, [1]) and transformer encoder features (`c-att-cat-tfenc`).

2) Varying Intrinsics During Inference: The proposed correlation layer allows dynamic adjustment to different camera intrinsics (i.e., stereo baseline and the horizontal focal length). To empirically validate this assumption two INSTR models are trained with 5,000 synthetic samples with rc_visard and Zed intrinsics, respectively. In both trainings, the maximum displacement is set to $d_{\text{max}} = 0.4$. Fig. 5 depicts evaluation on both test sensors and verifies the ability to adapt to untrained sensor intrinsics.

3) Single-RGB INSTR: As shown in Tab. IV, training INSTR without the disparity loss yields higher accuracy than single RGB-based predictions, indicating the local correlation itself as a strong cue - albeit not as informative as with guidance in the form of a designated disparity loss.

TABLE III: Architectural ablation study. All values denote mIoU [%] and are computed after 15 epochs of training. Values in **bold** denote the best results.

Ax. Bl.	Aux. loss	Query Proc.	val	rc_visard	Zed
✗	✗	c-att-exp	57.55	62.66	56.58
✗	✓	c-att-exp	61.35	67.48	62.53
✓	✗	c-att-exp	60.08	63.41	60.08
✓	✓	c-att	69.68	69.12	68.20
✓	✓	c-att-cat-bb	69.30	68.33	63.47
✓	✓	c-att-cat-tfenc	68.16	69.85	65.28
✓	✓	**c-att-exp**	**71.28**	**70.35**	**67.74**

4) Depth Evaluation: As mentioned, INSTR additionally predicts a pixel-wise disparity map. Although this map only fulfils the task of auxiliary guidance, for completeness the L1 and the RMS error of the predicted disparity compared to the ground truth obtained from the rc_visard with pattern are listed in Tab. V. We also list the performance of AANet [60], a dedicated stereo predictor, pretrained on Sceneflow [61]. Note that we only consider object ground truth regions and discard incomplete areas from calculation. In addition, we

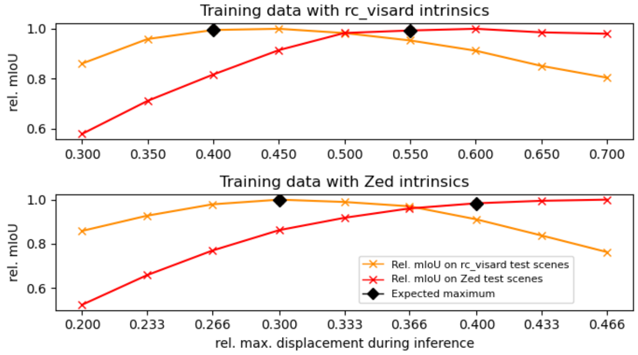


Fig. 5: Relative mIoU on test scenes recorded with the rc_visard (orange) and Zed (red) from training data with rc_visard (top) and Zed intrinsics (bottom). Our correlation layer with subpixel sampling enables generalization to novel sensor intrinsics.

TABLE IV: mIoU [%] on synthetic validation and our real world test set given different input modalities.

Input modality	Val	rc_visard	Zed
Single-RGB INSTR	69.30	66.58	63.52
INSTR w/o disp. loss	69.78	70.93	65.44
INSTR	77.46	74.92	74.31

experiment with utilizing the predicted depth of both [60] as well as INSTR as input for the depth-only version of [4], and receive 33.42 and 17.97 % mIoU on rc_visard scenes of STIOS.

5) Single-RGB INSTR on OCID Scenes: For completion, we evaluate our single RGB based approach on the OCID [52] dataset in Tab. VI alongside the RGB-based comparison on our test dataset (*Xiang et al. RGB* in Tab. II vs. *RGB* in Tab. IV). We only consider scenes from ARID10 and YCB10 to have less objects present than our upper bound of detectable objects (number of queries $N_q = 15$).

VI. Conclusion

We have proposed INSTR, a fast stereo-based instance segmentation approach (18fps) for unknown objects which addresses the issue of corrupted depth maps. By applying local horizontal correlation, the method is able to extract disparity-related as well as RGB-based features and learns a self-contained assessment of their significance. Furthermore, the correlation mechanism applies sub-pixel sampling, which enables dynamic adaption to the underlying camera parameters. Besides promising results on STIOS, we are able to grasp unseen objects as shown in Fig. 1 (and further in the video), and can segment a variety of object shapes/textures in completely different domains (see Fig. 6). Exploiting binocular image pairs, we hope to increase research interest towards robust stereo-aided robotic vision.

References

[1] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-End Object Detection with Transformers,” *arXiv:2005.12872 [cs]*, 2020.

TABLE V: L1 and RMS error [mm] evaluated on object regions of depth from rc_visard with pattern.

Method	L1 error	RMS error
Xu *et al.* [60]	09.73	24.00
INSTR	13.91	24.09

TABLE VI: mIoU [%] on two subsets (*ARID10* and *YCB10*) of the *OCID* [52] dataset.

Method	RGB	Depth	RGB+Depth
Xiang *et al.* [4]	34.71	76.76	80.91
Single-RGB INSTR	45.23	*n/a*	*n/a*

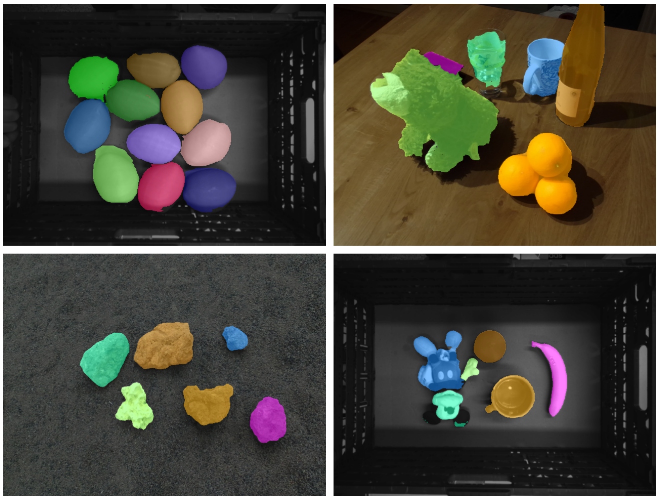


Fig. 6: Further arbitrary objects on surfaces in the wild. While being trained on random ShapeNet instances on synthetic tables, INSTR generalizes well to different domains: mangos in a box (top left), transparent objects (top right), and stones (bottom left). Though INSTR does not separate the stacked oranges (top right), it splits the Mickey Mouse (bottom right) into multiple instances. We hypothesize that this happens due to physical distance, texture and color properties of the objects.

[2] W. Friedl, H. Höppner, F. Schmidt, M. A. R. Garzon, and M. Grebenstein, “Clash: Compliant low cost antagonistic servo hands,” in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[3] S. Back, J. Kim, R. Kang, S. Choi, and K. Lee, “Segmenting Unseen Industrial Components In A Heavy Clutter Using RGB-D Fusion And Synthetic Data,” in *Int. Conf. on Image Processing (ICIP)*, 2020.

[4] Y. Xiang, C. Xie, A. Mousavian, and D. Fox, “Learning RGB-D Feature Embeddings for Unseen Object Instance Segmentation,” *arXiv:2007.15157 [cs]*, 2020.

[5] A. Valada, J. Vertens, A. Dhall, and W. Burgard, “Adapnet: Adaptive semantic segmentation in adverse environmental conditions,” in *Int. Conf. on Robotics and Automation (ICRA)*, 2017.

[6] B. J. Grzyb, L. B. Smith, and A. P. del Pobil, “Reaching for the unreachable: Reorganization of reaching with walking,” *Trans. on Autonomous Mental Development*, 2013.

[7] G. P. Bingham and M. A. Mon-Williams, “The dynamics of sensorimotor calibration in reaching-to-grasp movements,” *Journal of Neurophysiology*.

[8] Y. Zhang, J. Hare, and A. Prugel-Bennett, “Deep Set Prediction Networks,” in *Advances in Neural Information Proc. Systems*, 2019.

[9] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “Centernet: Keypoint triplets for object detection,” in *Int. Conf. on Computer Vision (ICCV)*, 2019.

[10] Z. Tian, C. Shen, H. Chen, and T. He, “Fcos: A simple and strong

anchor-free object detector," *Trans. on Pattern Analysis and Machine Intelligence*, 2020.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Int. Conf. on Computer Vision (ICCV)*.

[12] M. Bai and R. Urtasun, "Deep Watershed Transform for Instance Segmentation," in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[13] C. Xie, Y. Xiang, A. Mousavian, and D. Fox, "Unseen object instance segmentation for robotic environments," in *arXiv:2007.08073*, 2020.

[14] B. De Brabandere, D. Neven, and L. Van Gool, "Semantic Instance Segmentation with a Discriminative Loss Function," in *CVPR Workshops*, 2017.

[15] D. Neven, B. D. Brabandere, M. Proesmans, and L. Van Gool, "Instance Segmentation by Jointly Optimizing Spatial Embeddings and Clustering Bandwidth," in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[16] B. Romera-Paredes and P. H. S. Torr, "Recurrent Instance Segmentation," in *Eur. Conf. of Computer Vision (ECCV)*, 2016.

[17] A. Salvador, M. Bellver, V. Campos, M. Baradad, F. Marques, J. Torres, and X. Giro-i Nieto, "Recurrent Neural Networks for Semantic Instance Segmentation," *arXiv:1712.00617 [cs]*, 2019.

[18] K. Greff, R. L. Kaufman, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick, and A. Lerchner, "Multi-Object Representation Learning with Iterative Variational Inference," *arXiv:1903.00450 [cs, stat]*, 2020.

[19] C. P. Burgess, L. Matthey, N. Watters, R. Kabra, I. Higgins, M. Botvinick, and A. Lerchner, "MONet: Unsupervised Scene Decomposition and Representation," *arXiv:1901.11390 [cs, stat]*, 2019.

[20] M. Engelcke, A. R. Kosiorek, O. P. Jones, and I. Posner, "GENESIS: Generative Scene Inference and Sampling with Object-Centric Latent Representations," *arXiv:1907.13052 [cs, stat]*, 2020.

[21] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf, "Object-Centric Learning with Slot Attention," *arXiv:2006.15055 [cs, stat]*, 2020.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Int. Conf. on Neural Information Processing Systems*, 2017.

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words," *arXiv:2010.11929 [cs]*, 2020.

[24] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," *arXiv:2012.12877 [cs]*, 2021.

[25] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets V2: More Deformable, Better Results," in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[26] J. Liang, N. Homayounfar, W.-C. Ma, Y. Xiong, R. Hu, and R. Urtasun, "PolyTransform: Deep Polygon Transformer for Instance Segmentation," in *Conf. on Comp. Vision and Pattern Recog. (CVPR)*, 2019.

[27] T. Prangemeier, C. Reich, and H. Koeppl, "Attention-Based Transformers for Instance Segmentation of Cells in Microstructures," in *Int. Conf. on Bioinformatics and Biomedicine*, 2020.

[28] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia, "End-to-End Video Instance Segmentation with Transformers," *arXiv:2011.14503 [cs]*, 2020.

[29] E. Xie, W. Wang, W. Wang, P. Sun, H. Xu, D. Liang, and P. Luo, "Trans2Seg: Transparent Object Segmentation with Transformer," *arXiv:2101.08461 [cs]*, 2021.

[30] T. Pham, T.-T. Do, N. Sünderhauf, and I. Reid, "SceneCut: Joint Geometric and Object Segmentation for Indoor Scenes," *arXiv:1709.07158 [cs]*, 2018.

[31] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient Graph-Based Image Segmentation," *Int. Journal of Computer Vision*, 2004.

[32] A. Richtsfeld, T. Mörwald, J. Prankl, M. Zillich, and M. Vincze, "Segmentation of unknown objects in indoor environments," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012.

[33] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the Objectness of Image Windows," *Trans. on Pattern Analysis and Machine Intelligence*, 2012.

[34] A. Dave, P. Tokmakov, and D. Ramanan, "Towards Segmenting Anything That Moves," in *ICCV Workshops*, 2019.

[35] W. Boerdijk, M. Sundermeyer, M. Durner, and R. Triebel, "Self-Supervised Object-in-Gripper Segmentation from Robotic Motions," in *Conf. on Robotic Learning (CORL)*, 2020.

[36] P. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," in *Int. Conf. on Neural Information Proc. Systems*, 2015.

[37] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Eur. Conf. of Computer Vision (ECCV)*, 2014.

[38] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An information-rich 3d model repository," *arXiv:1512.03012 [cs]*, 2015.

[39] M. Denninger, M. Sundermeyer, D. Winkelbauer, Y. Zidan, D. Olefir, M. Elbadrawy, A. Lodhi, and H. Katam, "BlenderProc," *arXiv:1911.01911 [cs]*, 2019.

[40] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, "Segmenting Unknown 3D Objects from Real Depth Images using Mask R-CNN Trained on Synthetic Data," in *Int. Conf. of Robotics and Automation (ICRA)*, 2019.

[41] X. Cheng, Y. Zhong, M. Harandi, Y. Dai, X. Chang, T. Drummond, H. Li, and Z. Ge, "Hierarchical neural architecture search for deep stereo matching," *arXiv preprint arXiv:2010.13501*, 2020.

[42] M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. Sinha, "Object stereo - Joint stereo matching and object segmentation," in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[43] Ľ. Ladický, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. Torr, "Joint Optimisation for Object Class Segmentation and Dense Stereo Reconstruction," in *British Machine Vision Conference (BMVC)*, 2010.

[44] J. Zhang, K. A. Skinner, R. Vasudevan, and M. Johnson-Roberson, "DispSegNet: Leveraging Semantics for End-to-End Learning of Disparity Estimation From Stereo Imagery," *Robotics and Automation Letters*, 2019.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[46] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Int. Conf. on Computer Vision (ICCV)*, 2015.

[47] G. Yang, H. Zhao, J. Shi, Z. Deng, and J. Jia, "SegStereo: Exploiting Semantic Information for Disparity Estimation," in *Eur. Conf. of Computer Vision (ECCV)*, 2018.

[48] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv:1706.05587 [cs]*, 2017.

[49] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, and L.-C. Chen, "Axial-deeplab: Stand-alone axial-attention for panoptic segmentation," in *Eur. Conf. of Computer Vision (ECCV)*, 2020.

[50] H. W. Kuhn and B. Yaw, "The hungarian method for the assignment problem," *Naval Res. Logist. Quart*, 1955.

[51] K. C. L. Wong, M. Moradi, H. Tang, and T. Syeda-Mahmood, "3D Segmentation with Exponential Logarithmic Loss for Highly Unbalanced Object Sizes," in *Medical Image Computing and Computer Assisted Intervention*, 2018.

[52] M. Suchi, T. Patten, D. Fischinger, and M. Vincze, "Easylabel: a semi-automatic pixel-wise object annotation tool for creating robotic rgb-d datasets," in *Int. Conf. on Robotics and Automation (ICRA)*, 2019.

[53] C. Rennie, R. Shome, K. E. Bekris, and A. F. De Souza, "A dataset for improved rgbd-based object detection and pose estimation for warehouse pick-and-place," *Robotics and Automation Letters*, 2016.

[54] Z. Sui, Z. Ye, and O. C. Jenkins, "Never mind the bounding boxes, here's the sand filters," *arXiv:1808.04969 [cs]*, 2018.

[55] A. Ecins, C. Fermüller, and Y. Aloimonos, "Cluttered scene segmentation using the symmetry constraint," in *Int. Conf. on Robotics and Automation (ICRA)*, 2016.

[56] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Yale-cmu-berkeley dataset for robotic manipulation research," *Int. Journal of Robotics Research*, 2017.

[57] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[58] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," *arXiv:1711.05101 [cs, math]*, 2019.

[59] T. Hodaň, M. Sundermeyer, B. Drost, Y. Labbé, E. Brachmann, F. Michel, C. Rother, and J. Matas, "BOP challenge 2020 on 6d object localization," in *ECCV Workshops*, 2020.

[60] H. Xu and J. Zhang, "Aanet: Adaptive aggregation network for efficient stereo matching," in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[61] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.