3D SCENE RECONSTRUCTION FROM RGB IMAGES

Răzvan-Paul ROTARU[1], Alexandru GRĂDINARU[2], Florica MOLDOVEANU[3]

Object recognition is significantly improving, allowing us to better understand and extract information from images. This paper presents a novel method for 3D scene reconstruction using a single RGB image, based on a known 3D model database. We use every detected object in an image to further process its pose and a corresponding 3D model by leveraging existing datasets of both 3D models and labeled images to understand and simulate perception correctly. This method produces a clean and lightweight representation of a scene. State-of-the-art research, implementation details, and evaluation results are presented.

Keywords: 3D scene reconstruction, object detection, pose estimation

1. Introduction

Computer-vision techniques are becoming more complex every year. The continuous growth in this area leads to solutions with astonishing results in the fields of robotics, autonomous driving, and even content generation.

In a timespan of just 6 years, neural network models such as YOLO evolved to its fifth iteration [18] and Dall-E to its second [24] in just 2 years. These works highlight the importance of and focus on improvement in both object detection and content generation fields. Looking forward, research in these directions may be combined to reach the goal of reconstructing complex scenes organically and realistically. This question arose in 1963 when Lawrence Roberts [26] implemented a system that infers 3D scenes from plain RGB images using primitives for his Ph.D. thesis.

Since then, numerous deep learning architectures have been developed to represent 3D structures in different ways, which include voxels, point clouds, TSDF (Truncated Signed Distance Field) volumes, and meshes.

While these approaches have shown significant promise, the lack of training sets hindered them from achieving complete restorations. Current databases that can be used for training networks to predict the shape of an object from images consist mainly of isolated objects. Furthermore, recreating volumes from micro elements tends to generate geometry that may have a tendency towards noise and excessive tessellation, resulting in invalid and unnatural shapes.

[1] Faculty of Computer Science and Information Technology, The National University of Science and technology POLITEHNICA of Bucharest, Romania, e-mail: razvan_paul.rotaru@stud.acs.upb.ro
[2] Faculty of Computer Science and Information Technology, The National University of Science and technology POLITEHNICA of Bucharest, Romania, e-mail: alexandru.gradinaru@upb.ro
[3] Faculty of Computer Science and Information Technology, The National University of Science and technology POLITEHNICA of Bucharest, Romania, e-mail: florica.moldoveanu@upb.ro

However, most of the existing solutions for 3D reconstruction are focused on isolated objects, and, for realistic outputs, images from multiple angles are required. Moreover, the result of those algorithms is also deteriorated by the obstructed zone from the scene, the main consequence being the appearance of unnatural gaps in the shape of the reconstructed objects. Limitations of this kind make the predictions unsuitable for many applications.

In this paper, we propose a solution, to surpass some of these inconveniences, that will require a single image for an entire 3D reconstruction of the scene within the image. To reach the goal of scene reconstruction, a pipeline of 2D recognition for object detection and 3D reconstruction by leveraging preexisting meshes will be the foundation of a neural network architecture. Thus, we aim to align detected objects in an image to the viewport by mapping each detected object in an image to a mesh. By training, we learn to classify the objects and predict a specific model for it by comparing silhouette renders of models of the same type with the object in the region of interest. A finer pose prediction will be determined using the cosine distance to correctly align the shapes to the image. The training datasets are COCO and ShapeNet. However, the architecture allows the expansion of datasets only by adding meshes.

2. State of the art

To present the modern techniques related to this field, we will classify them into 2D object recognition techniques and 3D shape prediction techniques, which fundamentally address two distinct problems, thus having multiple specific solutions. Afterward, we will discuss another important aspect of the State of the Art, the Datasets created for this kind of work.

Most of the methods for 2D object recognition have similar outputs of a multi-layered deep neural network. The output usually consists of a labeled bounding box per object instance, with each identified object having a segmentation mask or a simplified bounding box centered around it, although the information used for prediction varies from one implementation to another [14, 19]. However, the most important element of these methods is not the architecture of the network but the datasets.

The State-of-the-Art methods for this field are Mask R-CNN [14] and Fast R-CNN [10], which maintain and take advantage of the ever-growing training database and output an additional segmentation mask for each input, beside the bounding boxes.

Another ground-breaking research in object and image segmentation is the Segment Anything (SAM) project [37], which has a model trained on over 1 billion masks on 11M images.

However, the architecture of our object detection network is inspired by version 3 of YOLO [25], which has the particularity of dividing the images and analyzing every outputted chunk to find regions of interest that may contain known classes of objects in a hierarchical manner.

In the 3D shape prediction, multiple solutions exist, that are completely different from one another, but they might be grouped into the following categories:

- Multi-View Reconstruction - commonly known as Photogrammetry, is covered by a broad line of work, starting from more classical techniques, such as using binocular stereo vision [13] to approaches that learn structure priors and constrain the output shape to be more natural [17, 3, 28], and even to ones that use more complex deep learning techniques [1]. However, this work will only focus on single-image reconstruction.
- Single-View Reconstruction - different approaches for shape prediction have been used in the last couple of years. As mentioned before, a usual course of action for this step is to train a deep network to predict the positioning of unit elements of an object.

Depending on the implementation, the definition of a unit element will vary from as accurate as a point [12, 5] to patches [23] and primitives [35, 27].

Other methods use signed distance functions for predictions [7, 20, 21, 36] that offer greater flexibility regarding the complexity of the given structure and better means to infer deep learning. Voxel grids are also a reliable solution for shape prediction [6, 16], in addition, they support multiple resolutions and can be hierarchically distributed through octrees. A post-processing step can also be integrated into the algorithms pipeline to convert the voxels into one or more meshes (Fig. 1) [9].

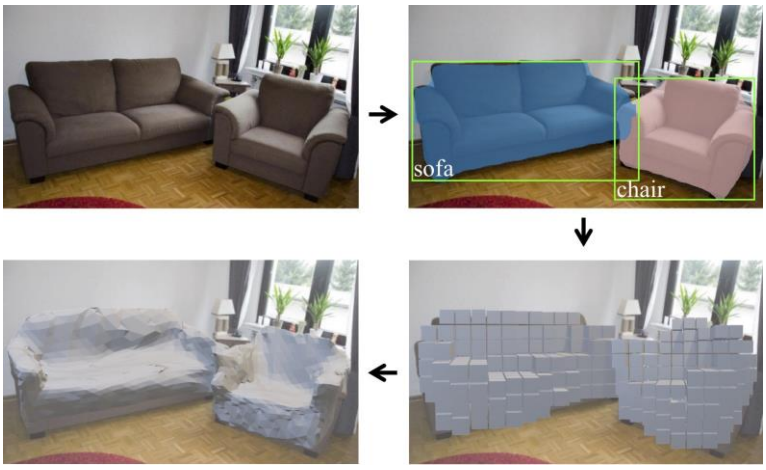

Fig. 1. Voxel grid conversion to mesh

However, because this work's goal is to reconstruct the objects in a scene in a more natural and precise way, a set of already existing models will be preferred instead of generated geometry. In this regard, algorithms have been predicting the orientation and pose of known shapes, but, mostly, are still limited to a single object instance.

Datasets represent another important aspect of the recent research in this field, which saw advances in 2D perception, through expanding already existing datasets, like COCO [31] and ImageNet [2], or by creating new ones, like ShapeNet [4] and Pix3D [33], which are large scale sets of CAD models and synthetically rendered images that can provide excellent training and testing data for field of study.

However, both the object identification and the CAD datasets still cover a small subset of object categories, that are not nearly enough to cover all the variance we have in captured images:

- ShapeNetCore has only 55 common object categories
- COCO – 80 object classes
- ImageNet – 1000 object classes

To improve the datasets, there is research on using generative methods to synthetically create labeled datasets or expand on the existing ones [30], as these techniques can be used with structured input data.

3. Proposed solution

We propose a solution that will handle organic end-to-end scene generation from a single image without depth information. To achieve these results, starting from a single RGB image, the method will need to be able to localize and distinguish objects by comparing the detected items of known classes to similar renderings of a 3D mesh set and select the best candidate, inferring their pose orientation to the image afterward. This approach learns to directly map each detected object in an image to a mesh in an end-to-end manner (Fig. 2).

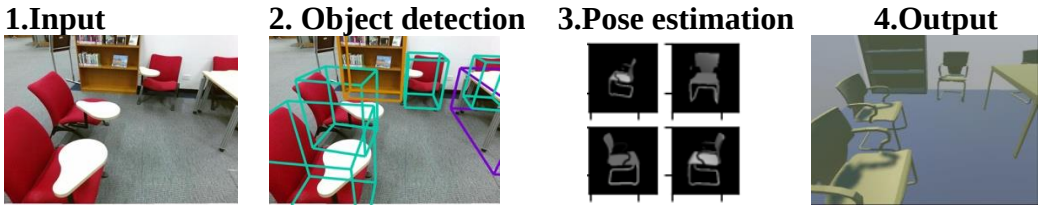

Fig. 2. Proposed solution

This approach draws inspiration from the success of Mask2CAD [32], Zhang et. al [36], and Nie et. al [21], which are novel techniques that manage to

reconstruct complex scenes that contain multiple shapes, by placing pre-existing 3D models of the identified objects within the scene. We also include the additional steps of Zhang et. al [36] of refining the final positions.

The architecture is composed of 3 networks:

- an initial layout estimation network (further referred to as LEN),
- an object detection network (further referred to as ODN)
- and a scene graph convolutional network (further referred to as SGCN).

More specifically, from the input image we will initially estimate the layout of the scene using a deep network (LEN) trained to identify the ceiling, walls, and floor, their depths, and rotations, which will be represented as a bounding box.

Afterwards, the image will be fed to the ODN to identify both semantic and structural aspects of the objects in the image. This network will output a coarse estimation of the identified object poses regarding their local implicit embeddings and the given layout from the previous network. As a result, in this step, a 3D bounding box and a class label will be predicted for all the objects detected.

In terms of object detection, the goal of this step would be to output a labeled mask for each object in the scene. Therefore, a simple solution for 2D Object Recognition is Detectron2 [8], a new system provided by Facebook based on the PyTorch3D [22] framework.

However, to accomplish this step a CNN following the architecture of YOLO [25] has been implemented, which proved to be accurate and fast. We trained the network on the COCO dataset [31] and learned to classify categories such as furniture, fruits, and electronics. The current accuracy of the model is somewhere above 91 percent. It is over-fitted on a couple of furniture categories, which have been used to develop and test the next branches. It should be noted that the model has not been trained for more than 12 hours due to resource limitations.

An alternative to the MS-COCO [31] dataset and the SUN RGB-D [29] dataset is creating a synthetic dataset in Unity, using the meshes available in ShapeNet [4] or other sources. This alternative has been explored, and a custom dataset has been created, using indoor assets from the Unity Asset Store. However, the training on this dataset proved not to be possible, as the synthetically generated data did not satisfy the condition of realism. The idea of this approach is learning to map between image views and different shape poses, resulting in an association between images and 3D geometry. Thus, driving inspiration from Mask2CAD [32] and IM2CAD [15] because 3D meshes and images belong to distinct categories, while an image is view-dependent, we facilitate the pose estimation and the construction of a shared space by rendering

K different views of every mesh. We chose *K = 20* in the experiments (Fig. 3). The resolution for each render is low due to performance limitations.

Because every prediction is class-labeled, each element in the joint space will be strongly class-related. As this step requires unsupervised learning, we surely know that even poor predictions maintain a correct structure.

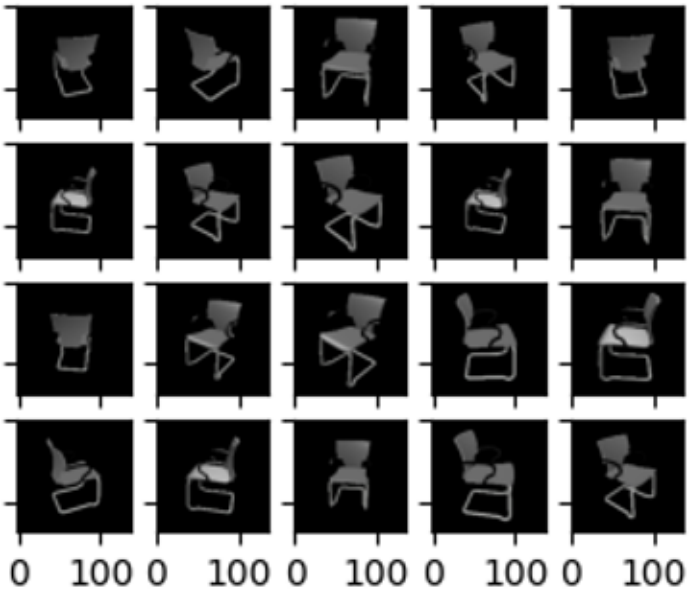

Fig. 3. A chair rendered from 20 different perspectives.

After the shared space is constructed, we can retrieve the shape of each prediction and apply the corresponding transformations to it. Following this structure, for an input image, at run-time, a scene containing only the detections should be outputted.

An argument can be made that a shared embedding space can be created to optimize this search, however, following IM2CAD [15], we use more than one layer to extract and compare feature vectors, which will make this optimization more irrelevant and, moreover, this would imply that each addition to the dataset must be encoded to this embedded space.

Regarding 3D shape prediction, this work aims to follow and improve Mask2CAD [32]. Because this method aims to reconstruct a more realistic and accurate scene than a geometry-accurate one, pre-existing meshes will be aligned to the viewport and not generated at the run-time.

For this objective, we trained a convolutional neural network on SUN RGB-D [29] and ShapeNet [4] to predict the 6 degrees of freedom (the 3D rotation and translation) of every item known in the scene. This process was eased by the 3D transformation API and differential renderer provided by PyTorch3D

[22] and the structure of the dataset which includes indoor scenes with labels and 3D bounding boxes.

Firstly, the object detection network will be trained to learn a 3D bounding box estimation from a labeled 2D bounding box, and afterward, the local implicit embeddings will be learned for each shape from contextual data.

As each model was rendered from *K* different perspectives, the translation and rotation have already been encoded and can be extracted from there. The best matching render, and automatically, the best matching model will be chosen by a Siamese Neural Network[4] trained to minimize the cosine distance between two images, the region of interest (RoI) of a detected object, and all renders of its class based on feature vectors of 7 different convolutional layers.

Another method that was considered was comparing the silhouettes by their Chamfer distance[5], however, the best matching model could not be selected by computing the Chamfer distance between the silhouette of the output RoI of an image and its corresponding model rendered silhouette because this would require the assumption that the dataset is infinitely large (i.e. contains any shape of any chair) which is not the case.

The pose refinement process is handled by the SGCN, which provides a better scene structure, no object overlaps, and better positions than previous attempts to satisfy this need.

The SGCN is a deep neural network trained to learn realistic placements of objects within a bounded area. This training has been achieved over the SUN RGB-D [29] dataset for objects of over 40 classes, following Zhang et al.'s work [36].

As a result of this step, the generated scene looks more elegant, with continuous, light, and clean geometry, meshes that don’t feel watery, no strange holes, and no setbacks due to inadequate illumination. This final step will provide us with a scene configuration file that can be imported into a graphics engine, for which we chose Unity as our preferred engine.

4. Evaluation and results

Because this technique falls into the same category of subjectivity and lack of metrics that other content-generative solutions fall in, we cannot present the mathematical accuracy of our work, apart from the run-time of our final application. We run a series of experiments consisting of 5 images and their respective run-time in minutes as follows (Table 1):

- Series 1 represents the run-time of the evaluation of an image containing 4 identified objects,

[4] https://en.wikipedia.org/wiki/Siamese_neural_network
[5] https://github.com/UM-ARM-Lab/Chamfer-Distance-API

- Series 2 of an image containing 8 identified objects,
- Series 3 - 4 identified objects,
- Series 4 - 7 identified objects,
- Series 5 - 6 identified objects.

Table 1

Run-time of different experiments (minutes)

Series	LEN	ODN	SGCN	Model selection
1	0.01	0.01	0.01	4.2
2	0.05	0.03	0.05	5
3	0.03	0.05	0.01	6.5
4	0.08	0.04	0.03	3
5	0.02	0.02	0.01	7

These tests have been measured on a system with an Intel i7 4700 CPU and a Nvidia 1050 GTX GPU.

By examining the run-time results, we could certainly state that the work of this paper could not sustain a real-time application scenario, as it currently requires more than 5 minutes for a scene generation and an additional 20 to 30 seconds for asset loading (measured in Unity). Further improvements are necessary for it to become a faster solution to the presented problem.

Moreover, we followed common practices and created a survey for individuals to evaluate our work based on how accurate they considered our final scene to be, how accurate the resemblance of the viewport was, and how accurate the models were (Fig 4).

Fig. 4. Survey results.

The results of 24 different persons showed that our method is satisfactory but could be improved in the future by extending the model dataset or improving the selection method.

Even if the viewport tends to be the strong point of our solution, another improvement that could be made is the object placement within the scene, which proved to be deceitful and mostly satisfies just the viewport of the image.

An evaluation that is impossible to make is the comparison between our solution and the previous one, because of our specific choice to use realistic pre-modeled meshes instead of generating meshes at run-time, which can be observed in the following images. The first example (Fig. 5) is an example of scene generation done in Total3DUnderstanding [21] used as a ground comparative result.

Fig. 5. A scene of an indoor table with 2 chairs reconstructed by Total3DUnderstaning [21].

We can observe in Fig. 6 that, even if the viewport looks correct, the object placement may not be entirely realistic when inspecting the scene closely.

Fig. 6. A scene of an indoor table with 2 chairs reconstructed by our method.

However, these undesired results are not very common. In comparison, Fig. 7 proves that our method can produce organic environments.

Fig. 7. A scene of an indoor office reconstructed by our method.

All in all, we demonstrated that our approach generates realistic high-quality scenes by combining several novel approaches in the literature and adding an additional step of using a high-definition realistic 3D models database that matches identified objects before rendering the scenes, providing a lightweight, clean, and compact reconstruction of the scene captured in the image.

6. Conclusions

This paper presents advancements in the generation of 3D scenes from images without depth information. After analyzing the current state-of-the-art techniques, a solution that improves on the existing methods was proposed. We presented in this paper some implementation details and results.

The comparison to other state-of-the-art techniques that address this problem cannot be easily realized. While novel approaches in this domain are published every year, not many are open-source, and we would be limited to comparing just a few results made public by the authors. The evaluation shows promising results, and we can state that we successfully addressed the issue of inconsistent geometry within the scene by leveraging already existing models and correlating them with objects of known classes, hence creating realistic environments that can be used for content generation in virtual reality scenarios. Doing so, we extended the astonishing results of previous techniques ([36] [21], [15] and [32]) in a different direction which we believe is more suitable for the target problem we address in this paper.

Further work is required to improve the speed performance, the model dataset, and the selection method.

Acknowledgement

The results presented in this article have been funded by the Ministry of Investments and European Projects through the Human Capital Sectoral

Operational Program 2014-2020, Contract no. 62461/03.06.2022, SMIS code 153735.

REFERENCES

[1] Christian Hane Abhishek Kar and Jitendra Malik. Learning a multi-view stereo machine. NeurIPS, 2017.
[2] Ilya Sutskever Alex Krizhevsky and Geoff Hinton. Imagenet classification with deep convolutional neural networks. NeurIPS, 2012.
[3] Victor A. Prisacariu Amaury Dame and Carl Y. Ren. Dense reconstruction using 3d object shape priors. CVPR, 2013.
[4] Thomas A. Funkhouser Angel X. Chang and Leonidas J. Guibas. Shapenet: An information-rich 3d model repository. CoRR, 2015.
[5] Chen Kong Chen-Hsuan Lin and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. AAAI, 2018.
[6] Danfei Xu Christopher B. Choy and JunYoung Gwak. A unified approach for single and multi-view 3d object reconstruction. ECCV, 2016.
[7] Maximilian Denninger and Rudolph Triebel. 3d scene reconstruction from a single view-port. ECCV, 2020.
[8] FAIR. Detectron2. https://github.com/facebookresearch/detectron2, accessed Feb 2023
[9] Jitendra Malik Georgia Gkioxari and Justin Johnson. Mesh R-CNN. ICCV, 2019.
[10] Ross B. Girshick. Fast R-CNN. ICCV, 2015.
[11] Frank R. Hampel. Introduction to Huber (1964) Robust Estimation of a Location Pa- rameter, pages 492–518. Springer, New York, 1992.
[12] Hao Su Haoqiang Fan and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. CVPR, 2017.
[13] Richard Hartley. Multiple view geometry in computer vision. Cambridge, 2003.
[14] K. He, G. Gkioxari, P. Doll´ar, and R. Girshick. Mask R-CNN. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 2980–2988, 2017.
[15] Hamid Izadinia, Qi Shan, and Steven M. Seitz. IM2CAD. CVPR, 2017.
[16] Chengkai Zhang Jiajun Wu and Tianfan Xue. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. NeurIPS, 2016.
[17] Xiuming Zhang Jiajun Wu, Chengkai Zhang and Zhoutong Zhang. Learning shape priors for single-view 3D completion and reconstruction. ECCV, 2018.
[18] Glenn Jocher, https://github.com/ultralytics/yolov5, accessed Feb 2023
[19] W. Kuo, A. Angelova, J. Malik, and T. Lin. Shapemask: Learning to segment novel objects by refining shape priors. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 9206–9215, 2019.
[20] Michael Oechsle Lars Mescheder and Michael Niemeyer. Occupancy networks: Learning 3d reconstruction in function space. CVPR, 2019.
[21] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian-Jun Zhang. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. CVPR, 2020.
[22] Jeremy Reizenstein Nikhila Ravi and David Novotny. Accelerating 3d deep learning with pytorch3d. CVPR, 2020.
[23] Yang Liu Peng-Shuai Wang, Chun-Yu Sun and Xin Tong. Adaptive O-CNN: a patch-based deep representation of 3D shapes. SIGGRAPH, 2018.
[24] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, arXIV, 2022.

[25] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. arXiv, 2018.
[26] Lawrence Roberts. Machine Perception of Three-Dimensional Solids. Garland Publishing, New York, ISBN: 0-8240-4427-4, 1963.
[27] Hao Su Shubham Tulsiani and Leonidas J. Guibas. Learning shape abstractions by assembling volumetric primitives. CVPR, 2017.
[28] Manmohan Chandraker Sid Yingze Bao and Yuanqing Lin. Dense object reconstruction with semantic priors. CVPR, 2013.
[29] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. Sun RGB-D: A RGB-D scene understanding benchmark suite. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 567–576, 2015.
[30] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[31] Michael Maire Tsung-Yi Lin and Serge Belongie. Microsoft coco: Common objects in context. ECCV, 2014.
[32] Tsung-Yi Lin Weicheng Kuo, Anelia Angelova and Angela Dai. Mask2cad: 3d shape prediction by learning to segment and retrieve. ECCV, 2020.
[33] Jiajun Wu Xingyuan Sun and Xiuming Zhang. Pix3d: Dataset and methods for single- image 3d shape modeling. CVPR, 2018.
[34] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph R-CNN for scene graph generation. ECCV, 2018.
[35] Andrew Luo Yonglong Tian and Xingyuan Sun. Learning to infer and execute 3d shape programs. ICLR, 2019.
[36] Cheng Zhang, Zhaopeng Cui, Yinda Zhang, Bing Zeng, Marc Pollefeys, and Shuaicheng Liu. Holistic 3d scene understanding from a single image with implicit representation. CVPR, 2021.
[37] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, Ross Girshick. Segment Anything, ICCV 2023.