Zero-shot virtual product placement in videos

Divya Bhargavi
Karan Sindwani
Sia Gholami
dbharga@amazon.com
ksindwan@amazon.com
gholami@amazon.com
Amazon Web Services
California, USA

ABSTRACT

Virtual Product Placement (VPP) is an advertising technique that digitally places branded objects into movie or TV show scenes. Despite being a billion-dollar industry, current ad rendering techniques are time-consuming, costly, and executed manually with the help of visual effects (VFX) artists. In this paper, we present a fully automated and generalized framework for placing 2D ads in any linear TV cooking show captured using a single-view camera with minimal camera movements. The framework detects empty spaces, understands the kitchen scene, handles occlusion, renders ambient lighting, and tracks ads. Our framework without requiring access to full video or production camera configuration reduces the time and cost associated with manual post-production ad rendering techniques, enabling brands to reach consumers seamlessly while preserving the continuity of their viewing experience.

CCS CONCEPTS

• **Computing methodologies → Computer vision**.

KEYWORDS

Object Detection, Image Segmentation, Virtual product insertion

ACM Reference Format:
Divya Bhargavi, Karan Sindwani, and Sia Gholami. 2023. Zero-shot virtual product placement in videos. In *ACM International Conference on Interactive Media Experiences (IMX '23), June 12–15, 2023, Nantes, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3573381.3597213

1 INTRODUCTION

This paper focuses on 2D ad placement (e.g. posters on a wall) rather than realistic virtual 3D ad objects (e.g. cans on a desk) due to several reasons. First, streaming platforms often purchase videos from third-party vendors who do not have access to camera parameters used for video production, making 3D scene understanding challenging. Second, there is no object/reference structure of known dimensions to calibrate scale for 2D to 3D point transformations. Lastly, live/real-time ad rendering applications with single-view cameras do not allow for the use of techniques like Structure from Motion (SfM) and multi-view stereo, as frames are processed sequentially [7, 8, 10, 20].

Existing computer vision-based VPP approaches are either semi-automatic [1], requiring user input for ad location, occlusion handling, and ad rendering adjustments, or automatic with ad replacement on specific targets such as billboards [24]. With the lack of standardized datasets and open-source repositories, the task of quickly prototyping an end-to-end solution for a potential commercial use is harder.

Our contributions are:

(1) Developing an end-to-end solution that automatically inserts 2-D ads into cooking show videos.
(2) Introducing 3 different ways of detecting empty spaces on indoor scene walls
(3) Building a framework that could generalize to 2-D ad insertions in any type of scene with minimal camera movement.

The rest of the paper is structured as follows: First, related previous work is summarized. Then, the details on implementing the zero-shot virtual product placement pipeline are provided. Next, the experiments are explained. And finally, the limitations and next steps, are presented.

2 RELATED WORK

2.1 Inverse Rendering in Indoor Scenes

Decomposing an RGB scene into material, geometry, and spatially-varying lighting has been widely studied for applications such as object placement and scene editing [15]. However, there is a lack of open-source implementations for commercial use and limited documentation on generalization across scenes within long-form videos. Challenges such as identifying empty spaces and tracking ad locations in an automated pipeline still need to be addressed.

2.2 Plane Detection

Identifying planar structures in scenes has been approached using Convolutional Neural Networks (CNNs). Existing models like PlaneNet and PlaneRecover [18, 19, 35] struggle to generalize to different scenes and smaller plane structures. PlaneRCNN improves on these issues by detecting planar regions and reconstructing a piecewise planar depth map from a single RGB image. However, it requires camera intrinsic parameters for refinement and 3D reconstruction. In our work, we utilize plane detection models to identify and delineate wall structures for empty space identification.

IMX '23, June 12–15, 2023, Nantes, France

ACM ISBN 979-8-4007-0028-6/23/06.
https://doi.org/10.1145/3573381.3597213

2.3 Instance Segmentation

Detecting and disambiguating distinct objects in an image has been approached using one-stage and two-stage models. Two-stage models generate segmentation maps by differentiating foreground and background [11, 16, 21] after identifying object proposals. One-stage methods [3, 28] can be anchor-based or anchor-free, and use parallel design and dense prediction networks. For our prototype, we chose to work with two-stage Mask-RCNN based backbones for their higher accuracy.

2.4 Light Estimation

Disambiguating properties of light, materials, and their interaction in 3D space is a sub-task in inverse rendering. While outdoor lighting settings are simplified, indoor settings require solving for complex spatially varying HDR light estimation [9, 29, 32]. In the absence of open-source pre-trained models, camera intrinsic, and stereo images, none of the deep learning methods apply to our use case. We use classical CV methods that learn global illumination properties and apply them to an ad image.

2.5 Key-point Detection and Description

Detecting stable interest points in an image and encoding them as descriptors is fundamental in various tasks such as SfM, SLAM, image matching, and vision localization. It has evolved from classical algorithms like SIFT and ORB [22, 26] to Transformer based models that capture global features through attention mechanisms [30]. We explore algorithms across all variations along with different key-point matching and homography estimation/outlier detection algorithms [5, 14] by comparing them using re-projection error

3 APPROACH

Our solution comprises 6 key steps (Figure 1), detailed in the following sections.

3.1 Identifying suitable placement location

The objective is to develop an ML model to identify suitable placement locations for 2D objects (posters, ad images) on a wall or other kitchen objects. Suitable placement locations can be on other kitchen objects as well (Microwave, oven, refrigerator) but these were not considered in the scope of this work. Models used for this task were selected based on them being state-of-the-art for a given task or doesn't require camera parameters. We experimented with 2 different strategies: One was a rule-based approach using pre-trained models while the other involved training a custom model on the data.

3.1.1 Rule-based approach. The rule-based approach involves sequentially executing the following steps:

(1) Detect walls using pre-trained panoptic-FPN segmentation [13] models in Detectron2 [33] library (see figure 2a). Additionally, detect distinct planar surfaces using PlanarReconstruction [36] model to disambiguate different folds of the wall (see figure figure 2b).
(2) Generate an empty space mask using the intersection of the results from wall segmentation and plane detection models (see figure 3a). The segmentation models/mask doesn't have the information to distinguish different blobs in the mask.
(3) Use a region proposal function to generate region bounding box for each blob(see Figure 3b). The rule-based pipeline returns suitable placement regions in an image but these regions may not be prospectively aligned.
(4) We align the bounding boxes by:
   (a) Using LETR(Line Segment Detection Using Transformers without Edges) model [34] to generate lines.(see figure 4a).
   (b) Classify these lines as vertical or horizontal by measuring slope.
   (c) Find the closest vertical and horizontal lines to align the region to wall line segments.
   (d) Calculate the distance between the center of the region and the endpoints on the line segment and take the pair with the minimum distance. (see figure 4b)
   (e) Compute adjusted region points with slope of LETR line segments. Given that a point (x_1, y_1) is at distance d away from (x, y). We can generate x_1 and y_1 co-ordinates using the following formulae.

$$r = \sqrt{1 + m^2} \tag{1}$$

$$(x_1, y_1) = (x + \frac{d}{r}, y + \frac{d.m}{r}) \tag{2}$$

3.1.2 Custom model approach. The rule-based approach may face latency and cascading error issues, so we also tested two custom modeling approaches: **Polygon Regression** and **Instance Segmentation approach**. The former uses Yolov5 [12] to predict a perspective-aligned bounding box, while the latter identifies patches/segments on the wall using a trained Mask-RCNN [11]. We evaluate both approaches based on the Intersection over Union (IoU) and angle deviation of the predicted bounding boxes from the ground truth.

3.2 Kitchen Scene Identification

It is a sub-task of the "Identifying empty space" objective. The VPP pipeline should be able to distinguish whether the frame being captured is within a kitchen or elsewhere and render the image accordingly. We use pre-trained CV models with a rule-based approach to classify a scene as a "kitchen scene" when a person is clearly visible (confidence scores above 0.95) and the surrounding area has kitchen-related artifacts (relevant shortlisted classes). We tested three pre-trained models, Amazon Rekognition (https://aws.amazon.com/rekognition/), Faster R-CNN [25], and RetinaNet [17], based on their SOTA accuracy on person and kitchen-related item classification metrics.

3.3 Occlusion Handling

In the absence of foreground-background maps and camera parameters, we formulated the 2-D VPP ad object to be on walls that are mostly in the background and it is reasonable to say any object that occludes its view will be on foreground. We only test the occlusion by humans as it is impossible to produce segmentation masks for unknown objects that the person in cooking shows might interact with. We benchmark semantic segmentation, instance segmentation, and panoptic segmentation models against

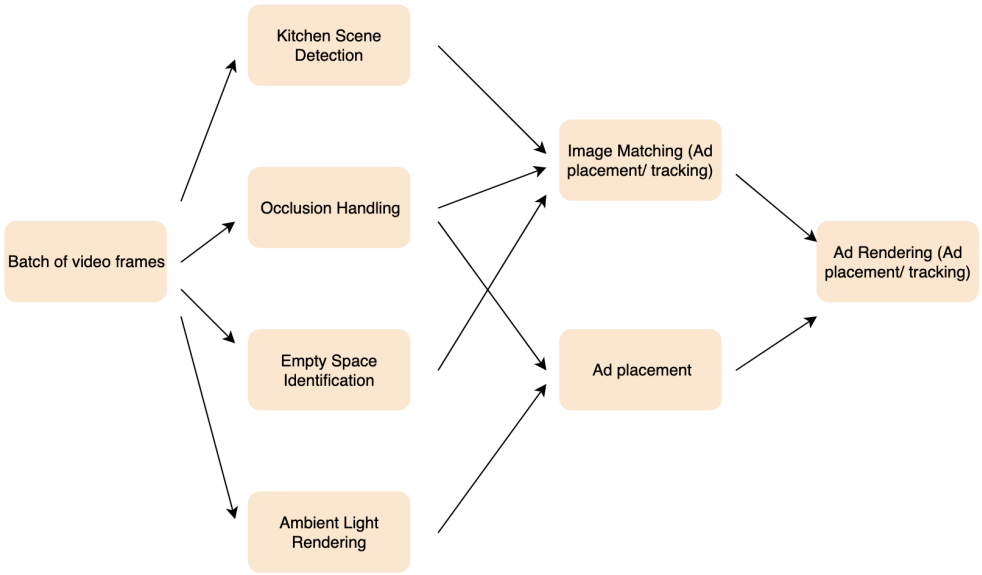

Figure 1: Automated Product Placement pipeline that processes a batch of frames. Kitchen scene, occlusion handling, empty space detection and ambient light are independent processes that understand context of scene at frame level. Image matching step occurs whenever a relevant scene is detected. It also utilizes image with occluding objects (like humans) removed so that the matching is mostly based on background cues. Ad placement steps carves out the portion of the ad image that is not occluded and is finally rendered on the empty space location.

(a) Wall detection

(b) Plane detection

Human Segmentation Data [27] that had high-definition masks of humans with different postures and backgrounds on IoU scores.

3.4 Ambient Light Rendering

The goal of this task is to match the perception of lighting of an advertisement image to a background image. We experimented with a method based on brightness calculation as presented in Szeliski, 2010 [31]. First, we calculate the brightness of the background image and then adjust the brightness of the ad to match that value.

$$g(x) = \alpha * f(x) + \beta \tag{3}$$

α and β are contrast and brightness respectively

3.5 Ad placement

The goal of this task is to place the ad image in the video, given its location coordinates and segmentation maps of occluding objects. To accomplish this, a computer vision module was developed that used OpenCV's [4] *getPerspectiveTransform* function to learn the transformation from the ad image to the placement location on the image. This method was tested as an alternative to simply pasting the image onto rectangular empty spaces. The transformation matrix warps the ad image according to the empty space

(a) Empty Space mask

(b) Region Proposals

(a) LETR output

(b) Lines closer to bounding box proposal

location dimensions. Before rendering the image, regions of the ad that are occluded by humans in the scene are masked out using segmentation maps.

3.6 Ad tracking

The objective for this task is to track the ad in a video for consistent and realistic rendering in the same location. We developed a computer vision module that tracks the location of ad in consecutive frames given previous video frame and its location coordinates. This module uses keypoint detector/descriptor, keypoint feature matcher and homography estimation functions. We have to note that this work was done on the premise that the camera parameters are unknown to learn 3D world to 2-D video frame mapping.

Tracking the location of Ad in consequent frames consists of the following tasks:

(1) Mask out occluding humans in image (refer to 3.3 task from above)
(2) Detection and Description: This involves understanding key features in an image and generating a feature-vector/ embedding. The models tested were the following:
   (a) Classical CV (OpenCV) : ORB, SIFT [22, 26]
   (b) Deep Learning: SuperPoint (implementation) [6], Kornia library [30].
(3) Remove features in and around occluding human. This is done so that the tracking is based background objects than the human features.
(4) Feature Matching: This involves matching the features generated in both the images for correspondence. We tested Brute Force, Single Nearest Neighbor, Mutual Nearest Neighbor, FGINN (1st geometrically inconsistent nearest neighbor ratio) [23] and GMS(Grid-based Motion) [2].
(5) Outlier Detection: This involves removing the outliers in feature matching using thresholds on "matching" metric. We tested RANSAC and MAGSAC [5, 14].
(6) Learn the homography matrix through OpenCV functions: This involves learning the transformation matrix (approximation of mild camera movement) between previous and current image.
(7) Get location coordinates: This involves applying the transformation on previous image Ad location coordinates to get new location coordinates for current image. We benchmark these algorithms using re-projection error.

4 EXPERIMENTAL RESULTS

4.1 Dataset

We used a dataset of 25 cooking shows in mp4 format videos with a resolution of 288x512 to develop an MVP for our automated framework. We sampled and labelled 1200 images (see figure 5). The video frames were labeled using the following mechanism:

(1) An image was labeled if it had a kitchen scene with empty space on walls and had the presence of a person in full view

Figure 5: Ground Truth Label

Table 1: Custom model results

Model	Approach	Avg IOU	Avg angle deviation	GT box overlap
Yolo-v5	Polygon Regression	0.56	3.27	40/42
Mask-RCNN	Instance Segmentation	0.52	3	37/42

(2) An image was not labeled/discarded if it was a close up of a cooking scene, a non-kitchen scene or the scene did not have any empty space on the wall
(3) An empty space was labeled using a polygon based bounding box
(4) 2-3 large empty spaces were marked for each image. Additionally, we augmented the labelled images with Gaussian Noise, Optical Distortion, Channel Shuffle and Random Cropping techniques.

4.2 Identifying suitable placement location

4.2.1 Rule-based approach. The rule-based approach (3.1.1) was evaluated based on qualitative results. Challenges were observed, including inconsistent results from wall detection, inaccurate results from PlanarReconstruction, sensitivity to camera movement, and dependency on wall background for alignment pipeline performance.

4.2.2 Custom model approach. Table 1 represents benchmarking of custom models for identifying suitable locations on our annotated dataset with respect to IoU (Intersection over Union) and Angle deviation between all 4 quadrilateral lines of ground truth and model predictions. Yolo-v5 (Polygon regression model) is relatively better at predicting empty spaces with low/ minimal overlap/occlusion with real-life objects. However, the Mask-RCNN (custom segmentation) model gave a lot more candidate spaces with lower deviation in perspective compared to Yolo-v5 on our ground truth. After qualitative (section 3.1.2) and quantitative analysis (table 1) of both models, we used instance segmentation approach to build the automated VPP pipeline.

Table 2: Scene classification results

Model	**Accuracy**
Retina-Net	0.926
Faster-RCNN	0.852
Amazon Rekognition	0.822

4.3 Kitchen Scene Classification

Kitchen scene is defined as a positive classification when a person is detected with a confidence of 90% or above and the image contains kitchen artifacts such as 'bottle', 'wine glass', 'cup', 'fork', 'knife', 'spoon' and 'bowl' with a confidence of 80%. The 90% threshold filters out cases where the person is partially visible or out of the scene. We used the same dataset as the empty space identification model, and RetinaNet had the highest accuracy in classifying kitchen scenes. This model is preferred for latency-related constraints due to its smaller architecture compared to Faster R-CNN.

4.4 Occlusion Handling

We quantitatively compare latency benchmarks and IoU results [27] of pre-trained Image segmentation models in table 3 and table 4 . We observed the following key takeaways: Semantic Segmentation models have better IoU performance than Panoptic and Instance

Table 3: Comparison of models on Human Segmentation dataset

Dataset/Framework	Segmentation Type	Model Name	IoU
COCO/Detectron2	Panoptic	panoptic_fpn_R_50_3x	0.907
COCO/Detectron2	Panoptic	panoptic_fpn_R_101_3x	0.908
COCO/Detectron2	Instance	mask_rcnn_R_101_FPN_3x	0.908
COCO/Detectron2	Instance	mask_rcnn_X_101_32x8d_FPN_3x	0.907
VOC/GluonCV	Semantic	fcn_resnet101	0.916
VOC/GluonCV	Semantic	psp_resnet101	0.920
VOC/GluonCV	Semantic	deeplab_resnet101	0.927
COCO/GluonCV	Semantic	fcn_resnet101	0.924
COCO/GluonCV	Semantic	psp_resnet101	0.927
COCO/GluonCV	Semantic	deeplab_resnet101	0.928
ADE/GluonCV	Semantic	fcn_resnet101	0.710
ADE/GluonCV	Semantic	psp_resnet101	0.716
ADE/GluonCV	Semantic	deeplab_resnet101	0.737

Table 4: Inference Time benchmark - CPU and GPU Latency

Method	Model	Image Size	CPU	GPU
Panoptic Segmentation	Panoptic fpn R50	2160 x 3840	7.497	0.178
		140 x 250	3.350	0.078
		281 x 500	3.521	0.080
		562 x 1000	3.598	0.085
	Panoptic fpn R5101	2160 x 3840	8.082	0.188
		140 x 250	4.148	0.090
		281 x 500	4.085	0.094
		562 x 1000	4.248	0.101
Instance Segmentation	Mask RCNN R50	2160 x 3840	4.831	0.165
		140 x 250	5.158	0.095
		281 x 500	4.977	0.097
		562 x 1000	4.985	0.103
	Mask RCNN R101	2160 x 3840	3.701	0.172
		140 x 250	3.620	0.082
		281 x 500	3.751	0.080
		562 x 1000	3.671	0.083
	Mask RCNN X101	2160 x 3840	6.206	0.202
		140 x 250	5.746	0.126
		281 x 500	5.859	0.128
		562 x 1000	5.744	0.131
FCN Semantic Segmentation	FCN ResNet101	2160 x 3840	94.25	2.610
		140 x 250	0.440	0.311
		281 x 500	1.250	0.458
		562 x 1000	5.928	0.424
PSP Semantic Segmentation	PSP ResNet101	2160 x 3840	94.800	2.063
		140 x 250	0.504	0.153
		281 x 500	1.414	0.080
		562 x 1000	6.367	0.155
DeepLab V3 Semantic Segmentation	DeepLab ResNet101	2160 x 3840	95.300	2.143
		140 x 250	0.447	0.095
		281 x 500	1.470	0.076
		562 x 1000	6.384	0.161

segmentation models. Instance/Panoptic Segmentation models performed 2x better in GPU/CPU inference latency than segmentation models. Models trained COCO, VOC dataset perform better in human segmentation than model trained over ADE dataset. Based on our qualitative evaluation on the dataset , we noticed Mask-RCNN model is unable to produce a prediction across all image resolutions and Panoptic segmentation models perform better than Mask-RCNN models across all image resolutions.

(a) Background Image

(b) Ad

(c) Brightness matching

Figure 6: Qualitative evaluation of ambient light rendering strategies.

4.5 Ambient Light Rendering

With lack of ground truth data and open source implements, we perform qualitative evaluation of the methods discussed in 3.4 as shown in figure 6

4.6 Ad placement

The rendering quality of the ad is affected by warping and interpolation techniques used by OpenCV. Higher-resolution images yield better rendering quality compared to lower-resolution images. Due to the lack of labeled data with rendered images, quantitative metrics for this task could not be tested. In some cases, resizing the ad image is necessary due to the identified empty space location being smaller than the original ad image dimension.

4.7 Ad tracking

The metric used was reprojection error which measures how far off pixel coordinates, the Ad location is on previous image with regards to its ground truth if we reverse the learned transformation. The metrics for top-2 feature matching algorithms are displayed in table 5. The lower the metric, better the pipeline is. There is no trend that deep learning models outperform classical techniques. While SuperPoint had the lowest error, Kornia had higher error than SIFT.

4.8 ML Pipeline

Our VPP pipeline is an automated python script that call multiple models hosted on 4 different GPUs tested on an Amazon EC2 p2.8xlarge instance. This pipeline currently has an 5-6 FPS (frames per second) for low resolution videos (288X512) and 1-2 FPS for high resolution (1080X1920) videos.

5 LIMITATIONS AND FUTURE WORK

We have identified the following areas for future exploration.

5.1 Identifying suitable placement location

For an accurate empty space detection model, we recommend a more comprehensive data annotation strategy that covers all possible empty spaces in a scene. Additionally, we would recommend training the model over multiple image resolutions and a larger annotated dataset for perspective-aligned predictions.

5.2 Kitchen-scene detection

To enhance the accuracy of the current rule-based method and mitigate false negatives, we recommend collecting labeled datasets that encompass various scenarios, such as different parts of the human body visible and indoor artifacts, to train models with high precision. This approach will improve the classification of scene semantics

5.3 Occlusion Handling

Virtual objects may flicker if the image segmentation is inconsistent. We recommend exploring image matting techniques and expanding occlusion detection to other objects in the kitchen scene to address this issue.

Table 5: Reprojection Error benchmark

Detection	Matching	Outlier filter	Reprojection error
Kornia LOTR	-	ransac	0.798
		magsac	0.786
Superpoint (Pytorch)	match_sym_fginn_intersection	ransac	0.755
	match_sym_fginn_intersection	magsac	0.759
Superpoint (Pytorch)	match_sym_fginn_union	ransac	0.748
	match_sym_fginn_union	magsac	0.753
SIFT (OpenCV)	match_sym_fginn_intersection	ransac	0.763
	match_sym_fginn_intersection	magsac	0.818
SIFT (OpenCV)	match_sym_fginn_union	ransac	0.803
	match_sym_fginn_union	magsac	0.795
Orb (OpenCV)	match_sym_fginn_intersection	ransac	0.754
	match_sym_fginn_intersection	magsac	0.793
Orb (OpenCV)	match_sym_fginn_union	ransac	0.756
	match_sym_fginn_union	magsac	0.816

5.4 Ambient Light Rendering

Since there are no publicly available datasets or ML models for benchmarking ad rendering, We recommend creating a curated dataset of labeled backgrounds, ads, and combined images that contain positive and negative samples to benchmark ad rendering. Additionally, we suggest experimenting with GAN architecture to create more realistic ads.

5.5 Ad placement

OpenCV-based methods use interpolation techniques that may cause a loss in resolution, particularly in low-resolution images. Quantitative benchmarks comparing CV-based rendering with high-definition rendering using software like Blend or Maya are currently unavailable. Realistic ad rendering also requires accurate placement of the ad at the correct scale and dimensions that are consistent with the 3D surroundings, which requires knowledge of camera depth and scale of known objects. Further research can be done by testing and evaluating VFX applications, exploring single view-based camera calibration, and depth estimation models for 3D scene understanding. However, due to the unavailability of camera depth and scale information in our dataset, we were unable to perform this evaluation.

5.6 Ad tracking

Our homography estimation-based tracking is an approximation for small camera movements. To improve tracking accuracy, we suggest exploring 3-D world-to-2D understanding using camera calibration. Benchmarking the effectiveness of tracking and realistic rendering by learning camera parameters using multi-view camera or single-view structure-from-motion algorithms on offline videos can help determine the best strategy for tracking.

6 CONCLUSION

In this paper, we present a solution for digitally placing a branded object into the scene of a movie or TV show. With our approach, advertisers can reach consumers without interrupting the viewing experience with a commercial break, as the products are seen in the background or as props. Our solution is easy to implement, requires minimal labeling, curation, supervision, and can be customized for various videos and advertisements. We hope the research community continue our work and develop better solutions for virtual product placement.

REFERENCES

[1] Ivan Bacher, Hossein Javidnia, Soumyabrata Dev, Rahul Agrahari, Murhaf Hossari, Matthew Nicholson, Clare Conran, Jian Tang, Peng Song, David Corrigan, et al. 2020. An advert creation system for 3D product placements. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 224–239.

[2] JiaWang Bian, Wen-Yan Lin, Yasuyuki Matsushita, Sai-Kit Yeung, Tan Dat Nguyen, and Ming-Ming Cheng. 2017. GMS: Grid-based Motion Statistics for Fast, Ultra-robust Feature Correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[3] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. 2019. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 9157–9166.

[4] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[5] Si-Yuan Cao, Jianxin Hu, Zehua Sheng, and Hui-Liang Shen. 2022. Iterative Deep Homography Estimation. *arXiv preprint arXiv:2203.15982* (2022).

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2018. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 224–236.

[7] Andrew W Fitzgibbon. 2001. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Vol. 1. IEEE, I–I.

[8] Yasutaka Furukawa and Carlos Hernández. 2015. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision* 9, 1-2 (2015), 1–148.

[9] Marc-André Gardner, Yannick Hold-Geoffroy, Kalyan Sunkavalli, Christian Gagné, and Jean-François Lalonde. 2019. Deep parametric indoor lighting estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7175–7183.

[10] Richard I Hartley. 1994. An algorithm for self calibration from several views. In *Cvpr*, Vol. 94. Citeseer, 908–912.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.

[12] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, TaoXie, Jiacong Fang, imyhxy, Kalen Michael, Lorna, Abhiram V, Diego Montes, Jebastin Nadar, Laughing, tkianai, yxNONG, Piotr Skalski, Zhiqiang Wang, Adam Hogan, Cristi Fati, Lorenzo Mammana, AlexWang1900,

Deep Patel, Ding Yiwei, Felix You, Jan Hajek, Laurentiu Diaconu, and Mai Thanh Minh. 2022. *ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference*. https://doi.org/10.5281/zenodo.6222936

[13] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. 2019. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6399–6408.

[14] Hoang Le, Feng Liu, Shu Zhang, and Aseem Agarwala. 2020. Deep homography estimation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7652–7661.

[15] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. 2020. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2475–2484.

[16] Justin Liang, Namdar Homayounfar, Wei-Chiu Ma, Yuwen Xiong, Rui Hu, and Raquel Urtasun. 2020. Polytransform: Deep polygon transformer for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9131–9140.

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[18] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. 2019. Planercnn: 3d plane detection and reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4450–4459.

[19] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. 2018. Planenet: Piece-wise planar reconstruction from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2579–2588.

[20] Sheng Liu, Xiaohan Nie, and Raffay Hamid. 2022. Depth-Guided Sparse Structure-from-Motion for Movies and TV Shows. *arXiv preprint arXiv:2204.02509* (2022).

[21] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. 2018. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8759–8768.

[22] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60, 2 (2004), 91–110.

[23] Dmytro Mishkin. 2019. matching-strategies-comparison. https://github.com/ducha-aiki/matching-strategies-comparison.

[24] Atul Nautiyal, Killian McCabe, Murhaf Hossari, Soumyabrata Dev, Matthew Nicholson, Clare Conran, Declan McKibben, Jian Tang, Wei Xu, and François Pitié. 2018. An advert creation system for next-gen publicity. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 663–667.

[25] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. https://doi.org/10.48550/ARXIV.1506.01497

[26] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. 2011. ORB: An efficient alternative to SIFT or SURF. In *2011 International conference on computer vision*. Ieee, 2564–2571.

[27] Vikram Shenoy. 2019. Human Segmentation Dataset. https://github.com/VikramShenoy97/Human-Segmentation-Dataset.

[28] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. 2019. Adaptis: Adaptive instance selection network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7355–7363.

[29] Pratul P Srinivasan, Ben Mildenhall, Matthew Tancik, Jonathan T Barron, Richard Tucker, and Noah Snavely. 2020. Lighthouse: Predicting lighting volumes for spatially-coherent illumination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8080–8089.

[30] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. 2021. LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8922–8931.

[31] Richard Szeliski. 2010. *Computer vision: algorithms and applications*. Springer Science & Business Media.

[32] Zian Wang, Jonah Philion, Sanja Fidler, and Jan Kautz. 2021. Learning indoor inverse rendering with 3d spatially-varying lighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 12538–12547.

[33] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. 2019. Detectron2. https://github.com/facebookresearch/detectron2.

[34] Yifan Xu, Weijian Xu, David Cheung, and Zhuowen Tu. 2021. Line segment detection using transformers without edges. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4257–4266.

[35] Fengting Yang and Zihan Zhou. 2018. Recovering 3d planes from a single image via convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 85–100.

[36] Zehao Yu, Jia Zheng, Dongze Lian, Zihan Zhou, and Shenghua Gao. 2019. Single-image piece-wise planar 3d reconstruction via associative embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1029–1037.