Diffusion Edge Detection Of Texture-less Objects

Matvey Ivanov
Automation and Control Institute (ACIN)
TU Wien
Wien, 1040
e11775774@student.tuwien.ac.at

Peter Hönig
Automation and Control Institute (ACIN)
TU Wien
Wien, 1040
hoenig@acin.tuwien.ac.at

Markus Vinzce
Automation and Control Institute (ACIN)
TU Wien
Wien, 1040
vincze@acin.tuwien.ac.at

Abstract

Edge detection of complex, smooth, transparent, reflective and texture-less objects is an unsolved problem in computer vision. In this work, an existing approach using diffusion in the image space is adapted to enable fast and accurate edge detection. The method is applied to texture-less industrial objects from the T-LESS and XYZIBD datasets. The models are trained on datasets, generated synthetically using BlenderProc. Three training datasets are created using T-LESS objects to evaluate the impact of edge type and object texturing on prediction quality. Two more datasets are generated using XYZIBD objects to investigate the influence of the crease angle used in edge rendering. The diffusion models are evaluated using the NMSE, SSIM, DICE, and CRISP metrics, to assess accuracy, structural fidelity, and perceptual sharpness. Experiments show that our approach achieves competitive edge prediction quality and consistently outperforms existing diffusion based methods in computational efficiency at a lower resolution, while offering overall better prediction fidelity compared to the Canny edge detector. With a runtime of 95ms per image on an NVIDIA RTX3090, the approach demonstrates suitability for deployment in robotic vision systems. A quantitative edge prediction quality evaluation is conducted on real-world test sets which are extended with the edge ground-truth.

1 Introduction

Edge detection in images remains a fundamental problem in computer vision [Huang and Huang, 2025], forming the basis for a wide range of tasks such as object detection, segmentation and scene understanding. Traditional techniques, such as Sobel [Kittler, 1983] and Canny [Canny, 1986], have been studied extensively [Kanopoulos et al., 1988], [Luo and Duraiswami, 2008], [Wang et al., 2021], but despite their efficiency, these methods are limited by their use of gradient-based features, which are sensitive to noise and variations in scene luminosity. These methods fall short in open-world robotics, where lightning conditions change constantly [Pulli et al., 2024].

To overcome the limitations of classical edge detectors, machine learning-based methods are introduced, as surveyed by [Hu, 2025]. Among these, diffusion models [Sohl-Dickstein et al., 2015] produce perceptually coherent and semantically rich edge maps by iteratively refining structural information, even under challenging conditions such as noise, occlusion, or low contrast. However, this denoising process introduces substantial computational overhead, hindering real-time applica-

The Third Austrian Symposium on AI, Robotics, and Vision (AIROV26).

bility. A recent approach by [Ye et al., 2024] employs a diffusion network with a U-Net [Ronneberger et al., 2015] backbone to achieve high precision, yet inference time remains a significant bottleneck.

Learning-based methods depend on reliable ground-truth annotations. In 2D imagery however, the absence of depth information often causes annotated edges to deviate from their true 3D locations. This limitation is mitigated by deriving edge ground truth directly from the corresponding 3D meshes. Numerous open-source 3D model repositories support such workflows. Datasets such as [Hodan et al., 2017], [Drost et al., 2017], [Kaskman et al., 2019], and [Xiang et al., 2018] primarily target industrial object scenarios, whereas [Morrison et al., 2020] provides a diverse set of geometries specifically designed for evaluating robotic grasping and manipulation.

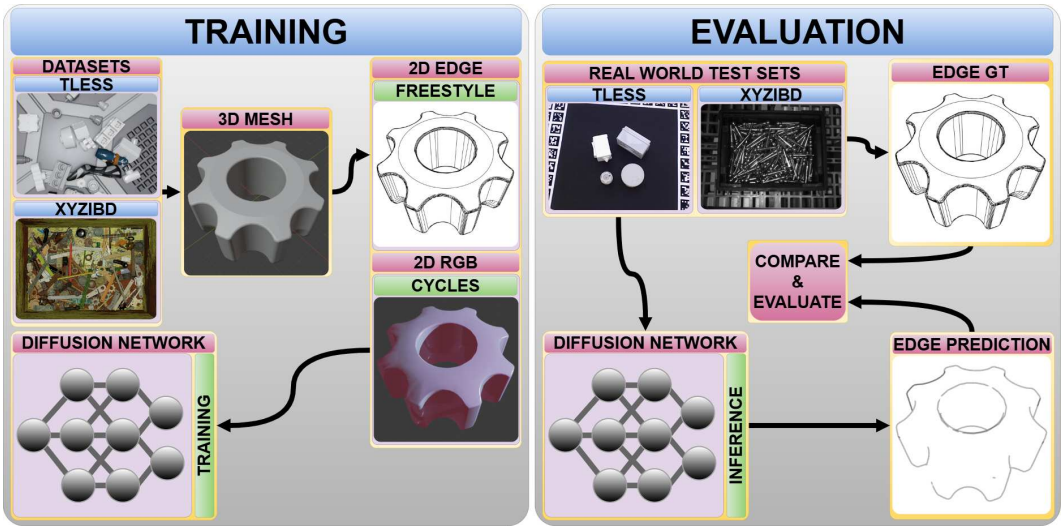


Figure 1: Complete Training-Evaluation Pipeline Block Diagram

This work presents a training and evaluation pipeline (see Figure 1) for diffusion based edge detection that maintains prediction quality while significantly reducing inference time. The objective is to facilitate the deployment of fast diffusion-based edge detectors on resource-constrained robotic platforms, where computational efficiency and reaction speed are critical.

We summarize our contributions as follows. We:

1. Extend the Blenderproc [Denninger et al., 2020] pipeline to allow precise, procedural edge rendering based on 3D-meshes and train a diffusion network to detect object edges in 2D, based on RGB images
2. Empirically prove that texture randomization improves prediction results [Hönig et al., 2025], when added to conventional domain randomization [Tobin et al., 2017]

2 Related Work

To highlight the constraints of existing methods related to edge detection, modern representative approaches and their trade-offs are reviewed. Works such as [Yu et al., 2017] and [Liu et al., 2017] integrate multi scale feature representations and semantic awareness to detect edges with high precision. Their performance is particularly strong in natural images. However, their reliance on deep backbones leads to high computational costs and long inference times. As a result, these methods are less suitable for resource constrained robotic systems [Ren et al., 2023]. A method targeting pose estimation refinement for transparent objects in laboratory settings via detection silhouette is proposed in [Weibel et al., 2026]. The approach remains constrained by the fidelity and confidence of silhouette extraction and by limited texture cues. It can be enhanced by training with large-scale synthetic datasets that provide precise per-object edge ground truth. Such datasets supply robust edge and silhouette supervision to support pose refinement. In addition, the fully synthetic TGF-Net

dataset for transparent objects [Yu et al., 2023] would similarly benefit from the inclusion of edge annotations, thereby strengthening its geometric supervision and improving downstream 6D pose estimation.

A key inspiration for this work is DiffusionEdge [Ye et al., 2024], which uses a diffusion probabilistic model with adaptive Fourier filters and a U-Net backbone to produce accurate edges in complex scenes, without requiring post-processing. Its effectiveness has been demonstrated in tasks such as 6D pose estimation of metallic objects [Leimeister, 2025]. However, the method suffers from long inference times on the order of magnitude of multiple seconds, making it impractical for real-time applications.

To address the run-time limitations imposed by prior methods, while retaining their robust edge detection performance in domain-specific contexts without reliance on manual annotation, the existing diffusion model architecture is trained on a set of synthetically generated datasets at a lower input resolution.

3 Synthetic Dataset Generation

Training a model to perform edge detection on real-world images requires the network to generalize well from its training set. To achieve this, domain randomization is used in all generated synthetic scenes, following the Benchmark for 6D Object Pose Estimation (BOP) convention. Different training datasets are rendered without textures and while applying object texture randomization, to further increase edge detection accuracy. Each scene is rendered from multiple camera viewpoints using BlenderProc [Denninger et al., 2020], which outputs RGB images, visibility masks, and edge maps for all selected target objects. Two types of datasets are generated, each based on different object collections and scene configurations:

1. T-LESS Floor Scenes With Distractors - CAD models from the T-LESS dataset [Hodan et al., 2017] are randomly placed on textured planes using physics simulation to ensure realistic spatial distribution. Distractor objects are sampled from HB [Kaskman et al., 2019], YCB-V [Xiang et al., 2018], and ITODD [Drost et al., 2017], representing industrial domain variability. Figure 2 illustrates an example of a T-LESS based scene without texture randomization, showcasing visible Freestyle edges and object masks.
2. XYZIBD Box Scenes - Multiple instances of a small subset of objects from the XYZIBD dataset [Huang et al., 2025] are dropped into a box using physics simulation. This setup reflects bin-picking scenarios with high object density and occlusion.

Figure 2: A Sample of a T-LESS Training Scene with Overlaid Target Object Edges on the Left and Visibility Masks on the Right

To obtain object-specific edge maps, the Blender Freestyle tool is integrated into the Blenderproc rendering pipeline [1]. Due to its CPU-bound nature, mesh simplification is performed to reduce computational overhead during edge rendering. This involves dissolving geometry below a specified angular threshold and segmenting non-planar faces to reduce mesh complexity. Additionally, loose vertices, edges, and faces are removed to eliminate redundant geometry using built-in operations available in the Blender Python module [2].

[1] https://github.com/DLR-RM/BlenderProc/pull/1203
[2] https://pypi.org/project/bpy/

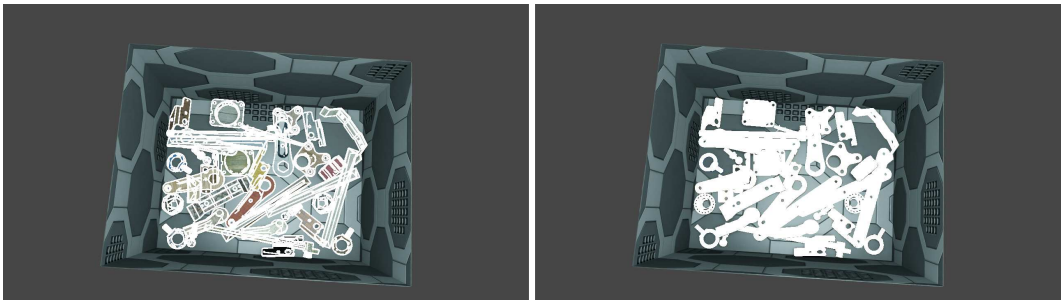
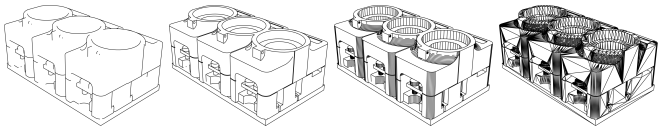


Figure 3: A Sample of a XYZIBD Training Scene with Overlaid Target Object Edges on the Left and Visibility Masks on the Right

Rendering performance is further improved by replacing the Cycles engine with EEVEE, which prioritizes rendering speed over photorealistic accuracy, while maintaining sufficient edge fidelity for the intended application. Freestyle provides a wide range of configuration parameters that influence the appearance and structure of the generated edge maps. In configuring Freestyle for line rendering, several parameters are enabled to ensure consistent and perceptually coherent edge extraction. Edge chaining is activated to guarantee that adjacent edge segments are properly connected, thereby producing continuous line structures. To preserve essential geometric and perceptual features, silhouette, crease and contour are enabled. These collectively ensure that major object outlines and sharp angular transitions are retained in the final render. In contrast, border edges are omitted to exclude outer object perimeters, while the visibility is set to only render visible edges. The angle between two adjacent faces, referred to as crease angle [3], plays a critical role in determining which mesh edges are rendered, as illustrated in Figure 4. A higher crease angle includes more shallow edges. A crease angle value of 180° (see Figure 4d) results in all mesh edges being drawn, whereas lower values like 175° (see Figure 4c), 160° (see Figure 4b) and 60° (see Figure 4a) exclude increasingly steep contours.



(a) 60° (b) 160° (c) 175° (d) 180°

Figure 4: Crease Angle Variation on a T-LESS Object

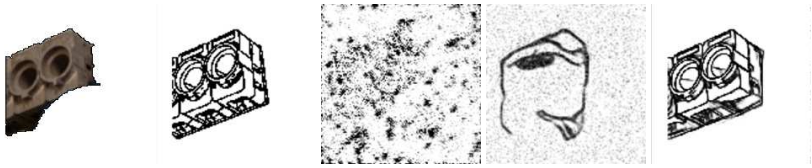
To evaluate the edge prediction quality of the trained models on real world images from T-LESS and XYZIBD, the test datasets are supplemented by the edge-ground truth. To achieve this, the scenes from the test sets are recreated in BlenderProc using available scene parameters and the ground-truth edge maps of each object in the scene are rendered using a crease angle of 160°. Predicted edges are later compared against the edge ground-truth in the test sets. Metrics aggregated in this manner enable a quantitative, real-world evaluation of the models trained on the fully synthetic training datasets. In this work, gray edges describe the grayscale edge maps provided by default by the Freestyle pipeline. These edge maps in the uint8 data format, contain values between 0 and 255 and represent the strength of an edge in each pixel. Clamped edge maps are created by setting all non-zero pixel values in the gray edge maps to 255. This results in all non-zero pixels being defined as an edge pixel, without regards to the strength of the edge. To evaluate the impact of edge type, object textures and crease angle on the edge prediction quality, multiple distinct training and test datasets are created. The same diffusion network architecture is trained separately on each dataset. In the clamped T-LESS dataset, texture-less objects are paired with clamped edge maps. The gray T-LESS dataset contains texture-less objects with grayscale edge maps. In the random texture T-LESS dataset, gray edge maps are used, while randomized textures are applied to the object surfaces. The XYZIBD training datasets are created using only gray edges, but with variation in texture, resulting

[3] https://docs.blender.org/manual/en/latest/render/freestyle/view_layer/line_style/modifiers/color/crease_angle.html

in two training sets. The XYZIBD test datasets are rendered using gray edges at six different crease angles.

4 Diffusion Model Training

In image generation tasks, U-Net commonly serves as the backbone denoising network [Rombach et al., 2022], learning to reverse the noise injection process iteratively. This property is especially advantageous for edge detection, where preserving fine-grained spatial features and their relations is critical. In this work, the U-Net operates at each denoising timestep to refine edge representations from noisy image samples. The training data generated by the BlenderProc pipeline provides synthetic RGB images along with corresponding object edges, masks, and visibility masks. The visibility masks are used to crop each image around the visible object, expand the bounding box by x1.5, and rescale it to a resolution of 128x128 pixels. Each training sample consists of three RGB channels and one grayscale edge channel, concatenated as input for the diffusion model. Each model is trained for approximately 120 hours on a single NVIDIA RTX 3090, using one of three dataset configurations: texture-less objects with clamped edges, texture-less objects with grayscale edges, and randomly textured objects with grayscale edges. During training, randomly selected training dataset samples are used as an estimate of the models edge detection ability. To achieve this, the model is switched to inference mode periodically and the edge predictions of the randomly selected samples are saved. The model architecture is based on the example from [4].



(a) (b) (c) (d) (e)

Figure 5: Running Validation During Model Training On Clamped Edges

Figure 5a shows the RGB and Figure 5b the edge ground-truth of a T-LESS validation sample. In early training stages (see Figure 5c), predictions exhibit high noise levels, with diffuse white regions loosely corresponding to the objects location. After approximately 2000 steps (see Figure 5d), the model begins to delineate object contours, though it continues to struggle with occlusion and complex geometries. By the end of training (see Figure 5e) the model consistently produces coherent and structurally plausible edge predictions. The models are deemed as sufficiently trained, when the training and validation loss converge to a minimum. For T-LESS-trained models 20 epochs are enough to achieve this. Models trained on XYZIBD training sets take 30 epochs to reach a comparable prediction quality.

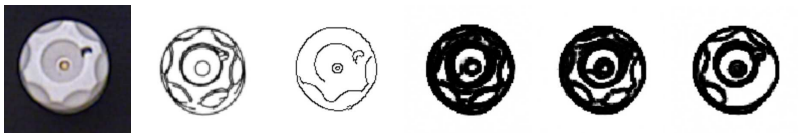
5 Model Quality Evaluation

The network predicts a single-channel edge map, which should ideally converge towards the edge ground-truth, independent of occlusion. However, when occlusion is severe and only a small portion of the object is visible, the network lacks sufficient context to infer object identity, orientation, or shape. In such cases, it tends to produce diffuse or spatially inconsistent edge responses, often concentrated around the region where the object is partially visible. These predictions reflect the model's uncertainty and its attempt to approximate the most likely object configuration under limited visual evidence.

Figure 6 provides a visual method comparison for a single sample from the T-LESS testset. The RGB image of the object from which all edge predictions are computed is shown on the left Figure 6a. It is followed by the edge ground-truth in Figure 6b acquired via Freestyle renderer. Figure 6c shows the Canny detector prediction with thin, but structurally incomplete edges. The remaining edges in Figure 6d, Figure 6e and Figure 6f stem from the models trained on clamped edges and texture-less

[4] https://huggingface.co/docs/diffusers/tutorials/basic_training#create-a-unet2dmodel

objects, gray edges and texture-less objects, as well as gray edges and randomly textured objects respectively. The model trained on clamped edges showcases thick, but mostly coherent edges. On the other hand the model trained on random textures does not fully capture the objects edges, but offers well defined edge contours. The gray edge, texture-less object model sits in between the two others, with relatively thin edges and a mostly complete edge structure.



(a) (b) (c) (d) (e) (f)

Figure 6: Edge Detection Method Comparison on a Single T-LESS Object

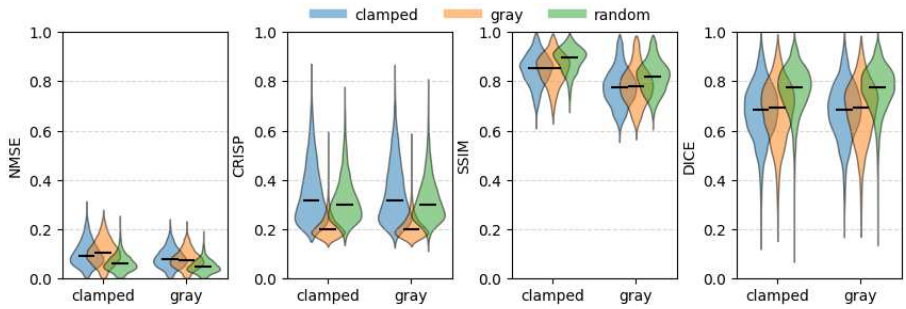


Figure 7: Edge prediction quality metrics NMSE↓, SSIM↑, DICE↑, CRISP↑ of models trained on clamped, gray, and random texture training datasets and evaluated on clamped and gray test datasets

To assess the overall performance of the trained models quantitatively, an evaluation is conducted using real world images from the T-LESS and XYZIDB test datasets, which are extended by the edge ground-truth via scene replication and the established Freestyle edge rendering approach. Predicted edge maps are compared against ground-truth edge maps using several established metrics, each capturing distinct aspects of edge quality, ranging from pixel-level accuracy to perceptual sharpness and structural fidelity. During evaluation, samples with a visibility mask containing fewer than 1% non-zero pixels when compared to the total number of pixels, are discarded. In these cases the target objects are deemed as too severely occluded.

Normalized Mean Square Error (NMSE) ([0,1] $\rightarrow$ lower is better $\downarrow$) provides a direct pixel-wise comparison between the predicted and ground-truth edge maps. This metric is sensitive to absolute intensity differences and does not consider spatial structure. Normalization is performed by scaling pixel intensities to the [0,1] range, enabling consistent comparison across images with varying brightness levels. Structural Similarity Index (SSIM) ([0,1] $\rightarrow$ higher is better $\uparrow$) [Wang et al., 2004] evaluates perceptual similarity between predicted and ground-truth edge maps by comparing local patterns of pixel intensities. It incorporates luminance, contrast, and structural alignment, and is computed over sliding windows, making it sensitive to localized distortions and edge deformation. In this work, SSIM is computed using a fixed intensity range of 255, consistent with 8-bit grayscale images. The Dice Similarity Coefficient (DICE/F-Score) ([0,1] $\rightarrow$ higher is better $\uparrow$) quantifies the spatial overlap between binary segmentation maps. Both predicted and ground-truth edge maps are thresholded to binary masks with values 0 or 255 and converted to boolean arrays. The Dice coefficient, equivalent to the F1 score in binary classification, measures the degree of overlap between the two masks, reflecting edge localization accuracy. The Crispness (CRISP) [Ye et al., 2023] ([0,1] $\rightarrow$ higher is better $\uparrow$) factor quantifies edge sharpness by computing the ratio of total pixel intensity after Non-Maximum Suppression (NMS) to that before NMS, using grayscale edge maps. A higher value indicates that NMS effectively preserves strong, localized edges while suppressing diffuse or thick contours. This metric penalizes blurred boundaries and favors well-defined edge structures.

Figure 7 presents violin plots of the discussed metrics for models trained on clamped edges and texture-less objects, gray edges and texture-less objects, and gray edges combined with randomly

textured objects. All objects stem from the T-LESS dataset. All three models are evaluated on two variations of the T-LESS test set. The first test set contains clamped and the second test set gray ground-truth edges. The models exhibit consistent performance across evaluation domains, indicating robustness to edge type, except in the case of SSIM, where the clamped test set provides a sharper edge ground-truth. The texture type used during training significantly influences prediction quality [Hönig et al., 2025].

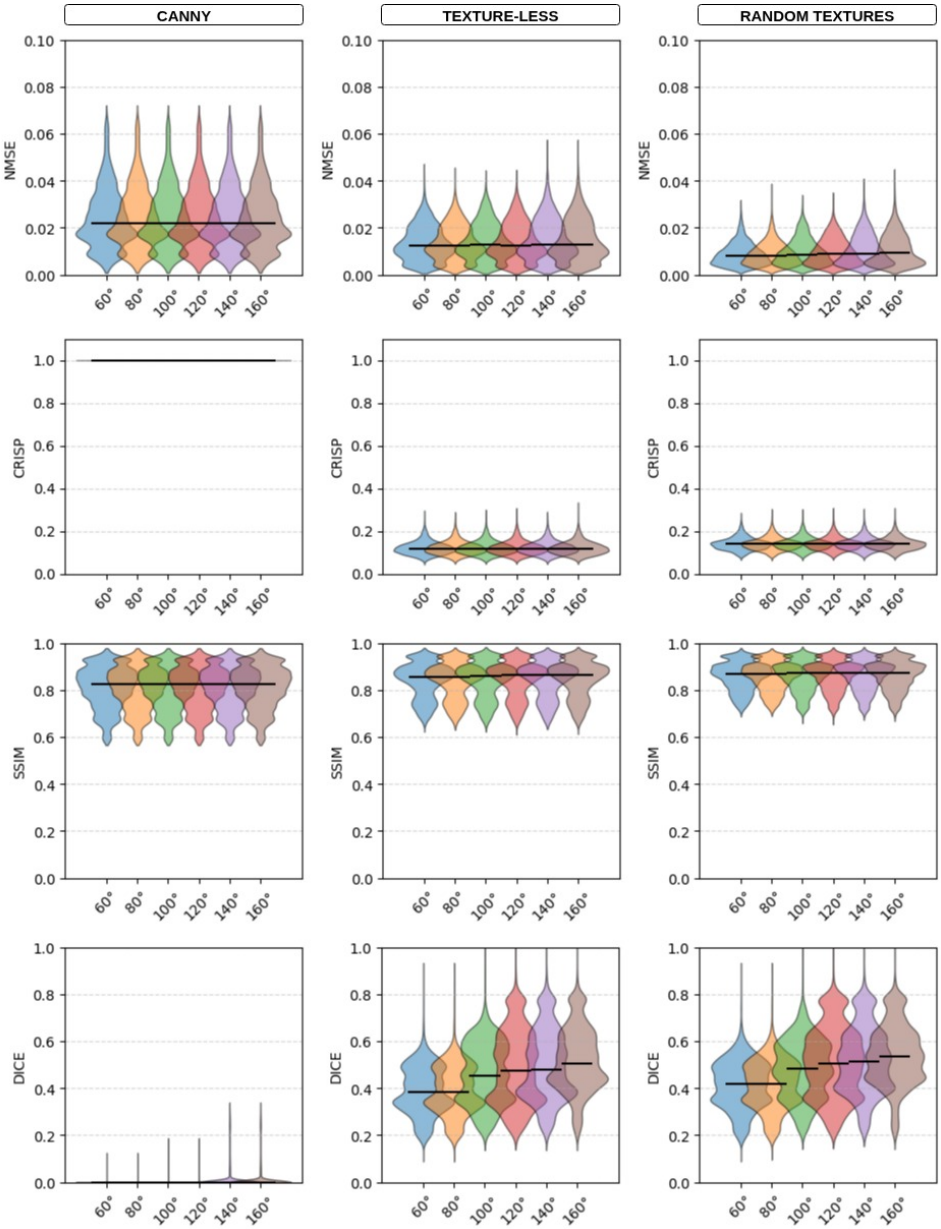


Figure 8: Edge Prediction Quality Metrics NMSE↓, SSIM↑, DICE↑, CRISP↑ for the canny edge detector, a texturelessly trained diffusion model and a diffusion model trained on objects with random textures

The model trained using gray edges and random textures consistently outperforms the clamped and gray models which are trained on texture-less objects across all metrics and test datasets, with the exception of the CRISP score. It yields the lowest NMSE, indicating better pixel-level accuracy, and achieves the highest SSIM, suggesting superior perceptual and structural similarity. It also yields the highest DICE coefficient, denoting improved segmentation overlap. Its CRISP score is slightly

worse than that of the clamped texture-less model, indicating sharp and well localized edges post-NMS. These results support the conclusion that the random texture model offers the most effective edge prediction performance for texture-less objects, aligning with the findings of As visualized in Figure 4, the crease angle has a large impact on the rendered edges. To evaluate its importance quantitatively, two diffusion models are trained on XYZ training datasets with a crease angle of 160° and gray edges. The first model is trained on a texture-less variant, while the textures in the other dataset are randomized. Both models are evaluated on the same XYZ test set with ground-truth edges rendered at crease angles: 60°, 80°, 100°, 120°, 140° and 160°.

The resulting metrics of both diffusion models are compared against the Canny edge detector [5] in Figure 8. The model trained on random textures performs better than the texture-less model across all metrics, while also being superior to Canny in NMSE, SSIM and DICE. Canny, by definition, exceeds in the CRISP metric, since NMS is part of its edge detection pipeline. This results in the CRISP score for Canny always being 1. On the other hand, Canny struggles with precise edge localization, especially at low crease angles, which results in a low DICE score across the validation dataset.

The trained models are evaluated on a system equipped with an AMD Ryzen 7 5800X 8-Core Processor and an NVIDIA RTX 3090 24GB GPU. Inference with diffusion models proceeds through a series of denoising steps, where the network iteratively refines a noisy input towards a clean edge prediction. Each step incrementally improves structural coherence and reduces noise, making the number of inference steps a key factor in both prediction quality and computational cost. For a 128×128 image, inference with 5 denoising steps takes approximately 95ms with a batch size of 1. In contrast, DiffusionEdge processes full scenes in a around 3.2 seconds. The step-wise nature of diffusion inference allows for fine-grained control over the quality–speed trade-off. To quantify the impact of inference step count on runtime, the clamped model is evaluated on the clamped test set using a batch size of 32. Increasing the number of inference steps from 1 to 2 results in a 1.3× increase in computation time, while increasing from 5 to 10 steps leads to a 1.7× increase. Empirically, 5 inference steps offer a favorable balance, where they allow for sufficient noise suppression and coherent edge maps production, while keeping inference time low. Increasing to 10 steps yields only marginal qualitative improvements.

6 Conclusion

This work presents a diffusion-based model for edge prediction of industrial, texture-less objects, trained on procedurally generated synthetic data and evaluated on real-world scenes. The resulting model demonstrates strong qualitative performance, particularly in handling complex geometries and partial occlusions, where it produces coherent edge maps even under limited visual evidence. Compared to existing model-based methods, the proposed approach offers improved computational efficiency at lower resolutions, making it suitable for deployment on resource-constrained platforms. Future work will focus on extending the method to multi-object and cluttered scenes, validating performance on broader real-world datasets, and integrating the model into robotic perception pipelines.

Declaration of AI-assisted Tools Used in Manuscript Preparation

Generative AI tools, including ChatGPT and Microsoft Copilot, were used to improve the clarity and readability of the manuscript. They were also employed to assist in writing Python code for implementation and experiment visualization. All generated material was reviewed and edited by the authors. The authors take full responsibility for the final content of the article.

Acknowledgments and Disclosure of Funding

This project is funded by the FFG, project GemSort, project number FO999923008 (www.ffg.at), the Austrian Science Fund (FWF), under project No. I 6114, project iChores, and by the EU program EC Horizon 2020 for Research and Innovation.

[5] https://docs.opencv.org/3.4/da/d22/tutorial_py_canny.html

References

John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Dmitry Olefir, Tomas Hodan, Youssef Zidan, Mohamad Elbadrawy, Markus Knauer, Harinandan Katam, and Ahsan Lodhi. Blenderproc: Reducing the reality gap with photorealistic rendering. In *16th Robotics: Science and Systems (RSS), Workshops*, 2020.

Bertram Drost, Markus Ulrich, Paul Bergmann, Philipp Hartinger, and Carsten Steger. Introducing mvtec itodd - a dataset for 3d object recognition in industry. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, 2017.

Tomáš Hodan, Pavel Haluza, Štěpán Obdržálek, Jiří Matas, Manolis Lourakis, and Xenophon Zabulis. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

Gang Hu. A mathematical survey of image deep edge detection algorithms: From convolution to attention. *Mathematics*, 13(15), 2025. ISSN 2227-7390.

Junwen Huang, Jizhong Liang, Jiaqi Hu, Martin Sundermeyer, Peter KT Yu, Nassir Navab, and Benjamin Busam. Xyz-ibd: High-precision bin-picking dataset for object 6d pose estimation capturing real-world industrial complexity, 2025.

Qinyuan Huang and Jiaxiong Huang. Comprehensive review of edge and contour detection: from traditional methods to recent advances. In *Neural Computing and Applications*, pages 2175–2209, 2025.

Peter Hönig, Stefan Thalhammer, Jean-Baptiste Weibel, Matthias Hirschmanner, and Markus Vincze. Shape-biased texture agnostic representations for improved textureless and metallic object detection and 6d pose estimation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 8806–8815, 2025.

N. Kanopoulos, N. Vasanthavada, and R.L. Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of Solid-State Circuits*, 23(2):358–367, 1988.

Roman Kaskman, Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2019.

J. Kittler. On the accuracy of the sobel edge detector. *Image and Vision Computing*, 1(1):37–42, 1983.

Lukas Leimeister. Electriceye: Metallic object pose estimation. Diploma thesis, Technische Universität Wien, Vienna, Austria, 2025.

Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. Richer convolutional features for edge detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Yuancheng Luo and Ramani Duraiswami. Canny edge detection on nvidia cuda. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2008.

Douglas Morrison, Peter Corke, and Jürgen Leitner. Egad! an evolved grasping analysis dataset for diversity and reproducibility in robotic manipulation. *IEEE Robotics and Automation Letters*, 5 (3):4368–4375, 2020.

Hönig Pulli, Hirschmanner Thalhammer, and Vincze. Enhancing transparent object pose estimation: A fusion of gdr-net and edge detection. In *Proceedings of Austrian Symposium on AI, Robotics, and Vision 2024*, pages 355–363, 2024.

Wei-Qing Ren, Yu-Ben Qu, Chao Dong, Yu-Qian Jing, Hao Sun, Qi-Hui Wu, and Song Guo. A survey on collaborative dnn inference for edge intelligence. *Machine Intelligence Research*, 20 (3):370–395, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241. Springer, 2015.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, 2017.

Shigang Wang, Xianghua Liao, and Guoqiang Wu. Infrared image edge detection based on improved canny algorithm. In *2021 IEEE 3rd Eurasia Conference on IOT, Communication and Engineering (ECICE)*, pages 280–284, 2021.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Jean-Baptiste Weibel, Clemence Dubois, Negar Layegh Khavidaki, Saifeddine Aloui, Mathieu Grossard, Markus Vincze, and Andreas Holzinger. Silref: Joint visual silhouette and tactile pose optimization for transparent object manipulation. *IEEE Robotics and Automation Letters*, 11(3): 2490–2497, 2026.

Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *Robotics: Science and Systems Proceedings*, 2018.

Yunfan Ye, Renjiao Yi, Zhirui Gao, Zhiping Cai, and Kai Xu. Delving into crispness: Guided label refinement for crisp edge detection. *IEEE Transactions on Image Processing*, 32:4199–4211, 2023.

Yunfan Ye, Kai Xu, Yuhang Huang, Renjiao Yi, and Zhiping Cai. Diffusionedge: Diffusion probabilistic model for crisp edge detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(7):6675–6683, 2024.

Haixin Yu, Shoujie Li, Houde Liu, Chongkun Xia, Wenbo Ding, and Bin Liang. Tgf-net: Sim2real transparent object 6d pose estimation based on geometric fusion. *IEEE Robotics and Automation Letters*, 8(6):3868–3875, 2023.

Zhiding Yu, Chen Feng, Ming-Yu Liu, and Srikumar Ramalingam. Casenet: Deep category-aware semantic edge detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.