Online 3D Edge Reconstruction of Wiry Structures from Monocular Image Sequences

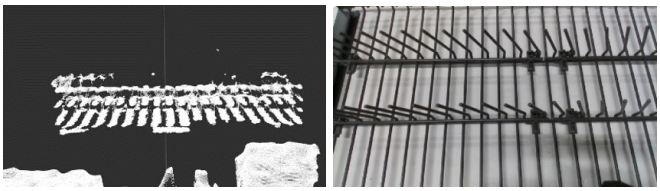
Hyelim Choi, Minji Lee, Jiseock Kang, and Dongjun Lee

***Abstract*— Three-dimensional (3D) reconstruction of wiry structures from vision suffers from thin geometry, lack of texture, and severe self-occlusions. We propose an online framework for reconstructing wiry structures whose skeletons are mainly straight as commonly found in man-made real-world objects in three dimensions (3D) from monocular image sequences. For an efficient and informative representation useful to address the harsh geometric nature of wiry objects (e.g., severe self-occlusion), we adopt a representation based on straight edges constructed from points. Specifically, we employ a robust maximum a posteriori (MAP) inference to construct sparse 3D points and subsequently use these sparse points to generate edge candidates whose beliefs are updated in a Bayesian manner. Then we take the set of 3D edges with beliefs greater than a threshold and apply a post-processing step to reject false edges. Experimental validation demonstrates the superior performance of our proposed framework in reconstructing 3D edges of wiry structures compared to existing state-of-the-art algorithms. We also demonstrate a manipulation task using the reconstruction that showcases the potential of the method to be easily used for subsequent robotic tasks.**

I. INTRODUCTION

Three-dimensional reconstruction is still challenging for wiry structures due to their thin geometry, lack of texture, and severe self-occlusion. Even commodity depth cameras cannot capture their geometry, as illustrated in Fig. 1.

An image-based approach is a promising solution, which can be categorized mainly into four classes with respect to the way each represents the 3D scene. First of all, point-based methods include most structure-from-motion (SfM) methods [1], [2], whose reconstructions are usually too sparse to represent low-textured wiry scenes. Although dense reconstruction methods (e.g., multi-view stereo (MVS) [3], [4]) can be exploited to produce a dense point cloud, they require intensive amount of computation and memory due to the little information points can accommodate. Second, lines can be exploited to render the scene in a compact yet informative manner. But works using lines [5]–[7] are not well-suited in cases of our interest (i.e., wiry structures) because the severe self-occlusion fragments line segments and damages the reconstruction. Third, curves can be used to depict more general scenes [8]–[11]. Nevertheless, these studies need segmentation [8], [9], and the computation is expensive because curves can only be handled with a set of points sampled on them. Lastly, deep learning-based methods can perceive 3D scenes from multi-view 2D images using neural networks [12]–[14], where the networks themselves are the implicit representations. However, such implicit representations pose a challenge in that we need an additional interpretation to use this for manipulation.



(a) (b)

Fig. 1. Snapshots of (a) a point cloud captured with RealSense D435i and (b) an image of IKEA Kungsfors dish drainer. The spokes are clearly visible in the image but they are not captured in the point cloud.

In this work, we propose a novel online strategy for reconstructing 3D edges of wiry structures from monocular image sequences. For an efficient and informative representation well-suited to address the challenges arising from such structures (e.g., severe self-occlusion), we adopt a representation based on straight edges constructed from points. More specifically, we employ a robust maximum a posteriori (MAP) inference to construct sparse 3D points and use these points to generate 3D edge candidates whose beliefs of being real physical 3D edges are updated in a Bayesian fashion. Then we take the set of edges with beliefs greater than a threshold and a post-processing is applied to filter out false edges. The proposed algorithm is experimentally validated on online reconstruction of real-world wiry objects, which demonstrates the superior performance of our framework in reconstructing 3D edges of wiry structures compared to state-of-the-art methods. This reconstruction can be efficiently utilized for a variety of robotic manipulation tasks, as demonstrated in a tableware manipulation.

II. METHODS

The overall architecture of our method is illustrated in Fig. 2. Our method is composed of two components: 1) sparse 3D point generation; and 2) edge inference. The first component generates sparse 3D points and the second one constructs edges from them.

A. Sparse 3D Point Generation

Using the feature correspondences obtained from the tracking, we formulate a maximum a posteriori (MAP) inference as a joint optimization to generate 3D points from the feature points and to correct the camera poses. Since

*This work was supported by Samsung Research.

H. Choi, M. Lee, J. Kang, and D. J. Lee are with the Department of Mechanical Engineering, IAMD and IOER, Seoul National University, Seoul 08826, South Korea (e-mail: helmchoi@snu.ac.kr; mingg8@snu.ac.kr; jskang0894@snu.ac.kr; djlee@snu.ac.kr).

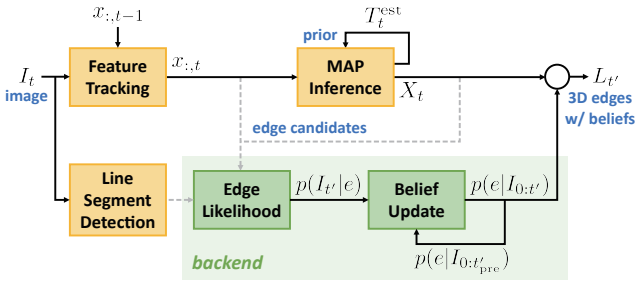


Fig. 2. Overall architecture of our proposed 3D edge reconstruction framework of wiry structures.

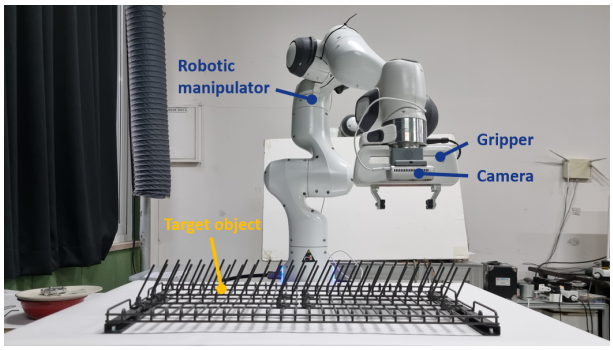


Fig. 3. The experiment setup. A camera (RealSense D435i) is mounted on a gripper (Franka Hand) attached to the end effector of the robotic manipulator (Franka Emika Panda). The target objects are put on the table.

self-occlusion creates many fake feature points, we utilize an expectation-maximization (EM) algorithm [15] for the MAP inference as a means of robust estimation. This is done by introducing a latent variable in the MAP formulation as

$$\Theta = \underset{\Theta}{\operatorname{argmax}}\, p(\Theta|x) = \underset{\Theta}{\operatorname{argmax}} \sum_{Z} p(Z, \Theta|x),$$

where $\Theta = [X, T]$, $X = \{X_i \in \mathbb{R}^3\}$ is the set of 3D points, $T = \{T_t \in SE(3)\}$ is the set of camera poses, t is a time step, $x = \{x_{it} \in \mathbb{R}^2\}$ is the set of feature points (x_{it} is the feature point at t corresponding to X_i), and $Z = \{Z_{it} \in \{0, 1\}\}$ is the set of latent variables (Z_{it} indicates whether x_{it} is a good feature to be included in the optimization).

B. Edge Inference

From the generated 3D points, we infer connecting edges using multiple images of different views since the appearance of connectivity from a single image can be misleading. We take current feature points and generate edge candidates by connecting them. The belief of each candidate is a conditional probability of being a real edge given observations and is updated via the Bayesian inference as

$$p(e|I_{0:t'}) = \frac{p(e|I_{0:t'_{\text{pre}}})p(I_{t'}|e)}{p(I_{t'}|I_{0:t'_{\text{pre}}})}$$

where $e \in \{0, 1\}$ is the indicator of being an edge, $I_{t'}$ is an image observation at t', t'_{pre} is the previous time step, $I_{t_1:t_2}$ is the image observations from t_1 to t_2, $p(e|I_{0:t'_{\text{pre}}})$ is the previous belief, $p(I_{t'}|e)$ is the likelihood of the observation at t', and $p(I_t|I_{0:t'_{\text{pre}}})$ is the normalizer ($= \sum_e p(e|I_{0:t'_{\text{pre}}})p(I_{t'}|e)$).

The likelihood of an edge candidate is evaluated by defining a metric that represents how much the segments in the neighborhood of the candidate suggest that it is a real edge (e.g., large when the segments are long and nearly parallel to the candidate). We would like to note that although this method of using line segments to evaluate the likelihood works in general settings, we can use any likelihood evaluation module that reasonably predicts the likelihood (e.g., neural networks working directly on images). At the end of the reconstruction process, we take the set of edges with beliefs greater than a threshold and a post-processing is applied to filter out false edges.

III. EXPERIMENTAL RESULTS

The experiment setup is shown in Fig. 3. The evaluations presented in this section are all done on a desktop with an AMD Ryzen 5 3600 6-core 3.59 [GHz] CPU, a 16 [GB] RAM, and an NVIDIA GeForce GTX 1660 GPU.

A. Comparative Evaluation

The proposed algorithm is evaluated in comparison with COLMAP [4], Line3D++ [5], EdgeGraph3D [10], and NeuS [13]. All algorithms are fed the same images and camera pose estimates from the robot measurements. Note that COLMAP uses a GPU while the other methods use a CPU only.

We present evaluation results on wiry objects and the results are depicted in Fig. 4. The reconstructions of COLMAP and Line3D++ have many outliers, with many of the lines of Line3D++ segmented into pieces, and EdgeGraph3D and NeuS suffer to depict the geometry in many cases. In contrast, the reconstruction using our framework provides models that mostly represent the object geometries with few outliers. The shortfalls of other algorithms can be attributed to several reasons: 1) thin geometry hinders the depth estimation of the structures; 2) line segments are fragmented into small pieces so that few meaningful reconstructions are made from them; 3) the detected line segments and curves are derived from the boundaries that change as the view changes, making the problem quite ill-posed; 4) a small number of images with a relatively small range of viewpoints are used; and 5) inaccurate camera poses without refinement deteriorate the reconstruction.

The number of parameters (i.e., 3#points for COLMAP and EdgeGraph3D, 6#lines for ours and Line3D++, and #SDF network parameters for NeuS) and the total execution times are presented in Tab. I. As can be seen in the table, the proposed framework reconstructs compact quantities of geometric primitives that articulate the object geometry and works online with a fast enough post-processing. We argue that running the reconstruction online is beneficial in that we can lower computation and memory requirements and access the reconstructed model anytime during the scanning, bringing an opportunity for active sensing.

TABLE I

THE NUMBER OF PARAMETERS AND TOTAL EXECUTION TIME (UNIT: [s]) FOR RECONSTRUCTION. THE EXECUTION TIME $x * y + z$ IN OURS MEANS THAT THE ALGORITHM IS RUN AT $1/x$[Hz] FOR y FRAMES AND THE POST-PROCESSING TAKES z[s].

	Ours		COLMAP		Line3D++		EdgeGraph3D		NeuS	
	# parm.	exec. time	# parm.	exec. time	# parm.	exec. time	# parm.	exec. time	# parm.	exec. time
Tetrahedron	102	$0.20 * 30 + 0.00$	93216	878.64	54	0.14	1074	3.38	529076	132604.12
Hexagonal prism	348	$0.20 * 30 + 0.00$	133572	846.36	54	0.17	2382	4.31	529076	135092.11
Dodecahedron	576	$0.20 * 30 + 0.00$	252753	898.80	384	0.51	9675	111.15	529076	135488.47
$2 \times 2 \times 2$ *cube*	6600	$0.25 * 30 + 0.11$	149826	867.96	324	0.38	4593	8.57	529076	136295.52
Wire Basket	5010	$0.33 * 50 + 0.03$	300843	1411.98	732	0.87	19419	77.31	529076	137613.54
Dish Drainer	16284	$0.33 * 50 + 0.53$	1694385	1358.70	3756	15.05	68016	307.90	529076	136386.25

TABLE II

SHAPE ESTIMATION ERROR (RMSE OF SAMPLED POINT CLOUD) OF EACH CASE (UNIT: [mm]). COLMAP, LINE3D++, AND EDGEGRAPH3D ARE ABBREVIATED AS CLMP, L3D+, AND EG3D RESPECTIVELY.

	Ours	CLMP	L3D+	EG3D	NeuS
Tetrahedron	**1.45**	4.04	1.97	7.63	-
Hexagonal prism	**2.31**	3.61	4.69	9.18	5.18
Dodecahedron	**2.77**	3.73	4.34	7.91	6.91
$2 \times 2 \times 2$ *cube*	**2.93**	4.53	5.62	6.41	6.51

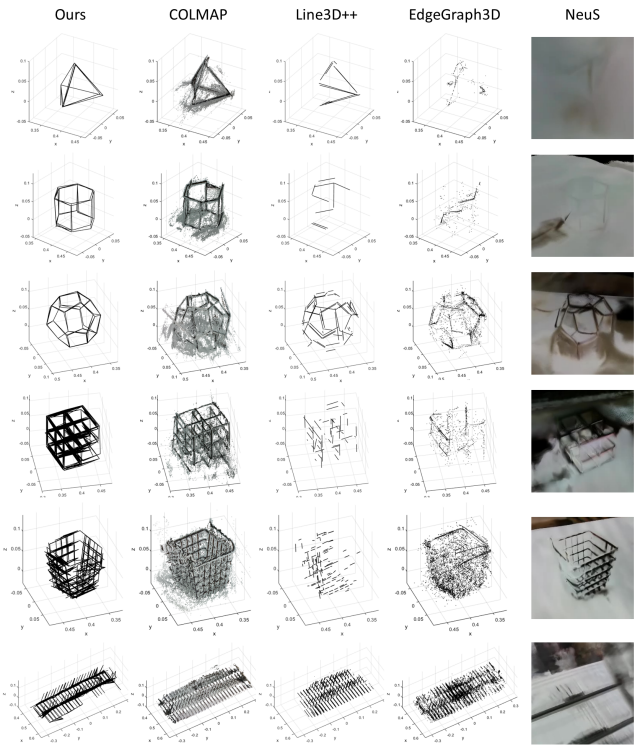


Fig. 4. Reconstruction results on wiry objects. The leftmost column contains the plots of the reconstruction using the proposed framework. The following columns contain the results using COLMAP [4], Line3D++ [5], EdgeGraph3D [10], NeuS [13], respectively. The points of COLMAP are colored using the image colors and the results of NeuS are rendered images given certain viewpoints. Rotating views are available at: `https://youtu.be/s1J9GVYt7Fs`

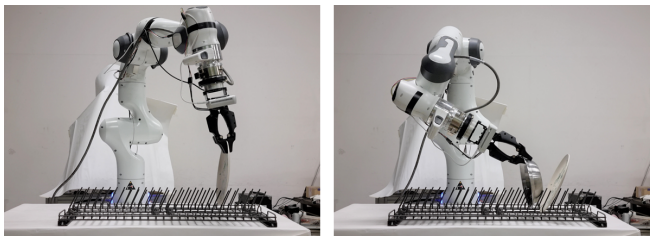


Fig. 5. Snapshots of a robotic tableware manipulation task. Two dishes are successfully placed in a row in the designated slots of the drainer.

We also evaluate the accuracy of the reconstruction by computing the distances after alignment between the points sampled from the reconstruction and the points sampled from the ground truth. Our framework achieves the smallest root mean square error (RMSE) values in all cases as presented in Tab. II, which implies that we obtain more accurate models with few outliers.

B. Manipulation Demonstration

We demonstrate how our framework can enhance perception for robotic manipulation tasks (e.g., [16]). A tableware manipulation task is performed, where a robot scans a dish drainer and places two dishes in their respective target slots on the wiry dish rack. For the demonstration, the robot is installed with a gripper and an FT sensor (ATI Gamma) for admittance control. We exploit the motion planner used in [17] to place dishes, and the experiment snapshots are presented in Fig. 5. This can only be possible with our framework which accurately estimates the configuration of the wire spokes of the rack and, consequently, the target slot for each dish.

IV. CONCLUSION

We propose an online 3D edge reconstruction framework that recovers scenes with wiry structures from monocular image sequences, which can be directly utilized in robotic tasks. For a compact and informative representation well-suited to wiry structures, we employ a robust MAP inference to construct sparse 3D points and use these points to generate edges whose beliefs are updated in a Bayesian way. The proposed framework is validated in experiments on the online reconstruction of wiry objects using a camera mounted on a robotic manipulator. The results demonstrate that our framework offers a more efficient and complete reconstruction compared to state-of-the-art approaches, and that it can be easily utilized for robotic manipulation.

REFERENCES

[1] C. Wu, "Towards linear-time incremental structure from motion," in *International Conference on 3D Vision (3DV)*, 2013, pp. 127–134.

[2] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4104–4113.

[3] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Towards internet-scale multi-view stereo," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1434–1441.

[4] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision*, 2016, pp. 501–518.

[5] M. Hofer, M. Maurer, and H. Bischof, "Efficient 3D scene abstraction using line segments," *Computer Vision and Image Understanding*, vol. 157, pp. 167–178, 2017.

[6] D. Wei, Y. Wan, Y. Zhang, X. Liu, B. Zhang, and X. Wang, "ELSR: Efficient line segment reconstruction with planes and points guidance," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 807–15 815.

[7] M. Hofer, A. Wendel, and H. Bischof, "Incremental line-based 3d reconstruction using geometric constraints." in *BMVC*, 2013.

[8] L. Liu, D. Ceylan, C. Lin, W. Wang, and N. J. Mitra, "Image-based reconstruction of wire art," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–11, 2017.

[9] P. Wang, L. Liu, N. Chen, H.-K. Chu, C. Theobalt, and W. Wang, "Vid2Curve: simultaneous camera motion estimation and thin structure reconstruction from an RGB video," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 132–1, 2020.

[10] A. Bignoli, A. Romanoni, M. Matteucci, and P. di Milano, "Multi-view stereo 3D edge reconstruction," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 867–875.

[11] S. Li, Y. Yao, T. Fang, and L. Quan, "Reconstructing thin structures of manifold surfaces by integrating spatial curves," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2887–2896.

[12] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *European Conference on Computer Vision*, 2020.

[13] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 171–27 183, 2021.

[14] L. Yen-Chen, P. Florence, J. T. Barron, T.-Y. Lin, A. Rodriguez, and P. Isola, "Nerf-supervision: Learning dense object descriptors from neural radiance fields," *arXiv preprint arXiv:2203.01913*, 2022.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodol.)*, vol. 39, no. 1, pp. 1–22, 1977.

[16] Z. Liu, W. Liu, Y. Qin, F. Xiang, M. Gou, S. Xin, M. A. Roa, B. Calli, H. Su, Y. Sun, and P. Tan, "OCRTOC: A cloud-based competition and benchmark for robotic grasping and manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 486–493, 2021.

[17] J. Lee, M. Lee, and D. J. Lee, "Uncertain pose estimation during contact tasks using differentiable contact features," in *Robotics: Science and Systems*, 2023.