2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Unsupervised 3D Part Decomposition via Leveraged Gaussian Splatting

Jae Goo Choy[1], Geonho Cha[2], Hogun Kee[3], and Songhwai Oh[3*]

***Abstract*— We propose a novel unsupervised method for motion-based 3D part decomposition of articulated objects using a single monocular video of a dynamic scene. In contrast to existing unsupervised methods relying on optical flow or tracking techniques, our approach addresses this problem without additional information by leveraging Gaussian splatting techniques. We generate a series of Gaussians from a monocular video and analyze their relationships to decompose the dynamic scene into motion-based parts. To decompose dynamic scenes consisting of articulated objects, we design an articulated deformation field suitable for the movement of articulated objects. And to effectively understand the relationships of Gaussians of different shapes, we propose a 3D reconstruction loss using 3D occupied voxel maps generated from the Gaussians. Experimental results demonstrate that our method outperforms existing approaches in terms of 3D part decomposition for articulated objects. More demos and code are available at https://choonsik93.github.io/artnerf/.**

I. Introduction

Understanding the 3D structure of a scene is one of the core problems in computer vision and graphics. Additionally, comprehending the 3D kinematics of articulated objects provides valuable information, especially in robotics tasks. Some researchers [1], [2], [3] incorporate 3D kinematic information of articulated objects to manipulate them. Therefore, inferring motion-based parts from RGB video is crucial, and unsupervised methods capable of inferring parts from unfamiliar objects are valuable for manipulating unseen articulated objects. Meanwhile, recent dynamic neural radiance fields (NeRF) methods enable us to reconstruct a 4D scene from an RGB video [4], [5], [6]. Kerbl et al. [7] proposed 3D Guassian splatting (3DGS), a novel view image synthesis technique representing the scene with 3D Gaussians. By utilizing differentiable splatting techniques [8] with Gaussians, it achieves not only high image synthesis quality but also a dramatic increase in rendering speed. 4D Guassian Splatting (4DGS) [9] applies the idea of 3DGS to enable novel view image synthesis in dynamic scenes, and like 3DGS, it offers high image synthesis quality and fast rendering speed.

WatchItMove [10] (WIM) is a method to infer a canonical space with 3D partitioned information of articulated objects from multi-view videos. WIM consists of two modules: a network that partitions the canonical space into a set of ellipsoids, and a network that predicts the pose changes of the ellipsoids over time. WIM successfully partitions the canonical space into parts unsupervisedly through image synthesis using the NeRF technique. However, WIM is vulnerable when a monocular video is given, and it subdivides objects into ellipsoids, leading to the disadvantage of unnecessarily dividing them into more parts than the actual number of parts.

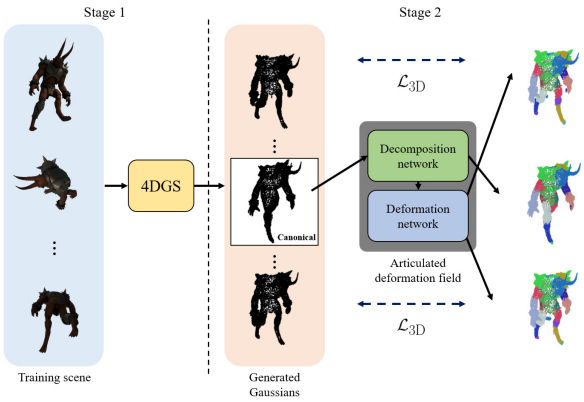

Fig. 1. Visualization of the entire process of the proposed method. In the first stage, Gaussians are generated by 4DGS, and in the second stage, the articulated deformation field is trained using the generated Gaussians to predict part segmentation and its deformation.

To resolve this issue, we propose a novel method to infer the 3D structure of articulated objects from a monocular video, leveraging Gaussian splatting techniques. In particular, we take three benefits of 4DGS: fast image rendering speed, fast training speed, and high quality Gaussians for dynamic scenes. The proposed method consists of two stages. In the first stage, 3D Gaussians are generated for each time stamp from the monocular video using 4DGS. In the second stage, an articulated deformation field is trained, which consists of a decomposition network that partitions the canonical space into parts and a deformation network that infers pose changes over time for each part. The whole process of the proposed method is visualized in Figure 1.

By using the articulated deformation field, we can deform the canonical 3D Gaussians to any time in the training scene. We train the articulated deformation field by rendering the images from the deformed 3D Gaussians and comparing those with the ground-truth images. However, this photometric loss alone is not sufficient to efficiently train the articulated deformation field, so we introduce a 3D reconstruction loss that takes into account the 3D shape of the Gaussians.

[1]J. Choy is with Sequor Robotics, Seoul 08376, Korea cjg429@gmail.com
[2]G. Cha is with NAVER Cloud, Seongnam 13561, Korea geonho.cha@navercorp.com
[3]H. Kee and S. Oh are with the Department of Electrical and Computer Engineering and ASRI, Seoul National University, Seoul 08826, Korea hogun.kee@rllab.snu.ac.kr, songhwai@snu.ac.kr
*Corresponding author

The 3D reconstruction loss is inspired from iterative closest point (ICP) techniques [11], [12], [13], [14]. In essence, when given two distinct point clouds, $\mathbf{X}$ and $\mathbf{Y}$, ICP aims to iteratively optimize a delta function $\Psi(\mathbf{X})$ by minimizing a distance metric, such as the chamfer distance [15], between the transformed point cloud $\mathbf{X}+\Psi(\mathbf{X})$ and $\mathbf{Y}$. To compute the chamfer distance between two point clouds, it is necessary to find the nearest point for each point in the point cloud, which can lead to instability during training. To address these challenges, the 3D reconstruction loss computes the distance between two Gaussians by voxelizing them.

Through extensive experiments, we demonstrate the effectiveness of the proposed method in unsupervised 3D part decomposition by comparing its performance on articulated objects datasets with other state-of-the-art methods. In summary, the contributions of this paper can be summarized as follows

- We propose a novel approach to infer the 3D parts of articulated objects from monocular videos leveraging Gaussian splatting techniques.
- Our method demonstrates the state-of-the-art performance in a 3D part decomposition task.

II. Related work

A. 3D part decomposition

The use of point clouds for 3D part decomposition is prevalent in the field, with various deep learning networks [16], [17], [18], [19], efficiently handling unordered point cloud information. Large-scale 3D part datasets like ShapeNetCore [20] and PartNet [21] have contributed to the success of these approaches. While these methods excel in decomposing parts from a single point cloud, they face challenges when encountering objects from unseen categories. To address this, several unsupervised methods use multiple point clouds and object motions for 3D part decomposition. For instance, Tzionas et al. [22] and Nunes et al. [23] utilize 3D motion tracking information for simultaneous 3D decomposition and mesh reconstruction of articulated objects, along with the recovery of 3D kinematics. In a similar vein, Choy et al. [24] employ point cloud registration [14] for motion-based 3D decomposition without the need for explicit 3D motion tracking. Additionally, some studies [25], [26], [27] propose motion-based part decomposition from images using optical flow. Notably, Noguchi et al. [10] introduce a method for decomposing articulated objects into parts from multi-view videos using the NeRF technique. Their approach is underpinned by the assumption that articulated objects can be effectively approximated by a set of ellipsoids.

B. Novel view image synthesis for dynamic scenes

Neural Radiance Fields (NeRF) is a method that encodes 3D information from various multi-view images and enables us to render images at arbitrary viewpoints based on this encoded information. Notably, NeRF techniques have evolved beyond static scenes, extending their applicability to dynamic scenes with an additional temporal dimension [4], [5], [6], [28]. The core concept underlying these advancements involves defining a time-invariant canonical space and use deformations to this space to achieve rendering across the entire temporal span. However, deviating from the mainstream, an alternative approach efficiently encodes high-dimensional 4D volumes using six planes. This alternative, as proposed in [6], enables the rendering of dynamic scenes without relying on a canonical space, presenting a novel perspective in the field of NeRF methods. 4DGS, on the other hand, uses a Gaussian splatting technique that maps 3D Gaussians to 2D, unlike the NERF method that calculates colors per ray, resulting in higher quality images and faster rendering speed.

III. Preliminary

A. 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) [7] is a computer graphics method for reconstructing a 3D scene from multi-view images. In this technique, the 3D scene is represented as a point cloud, where each point possesses Gaussian properties, centered at $\mathcal{X} \in \mathbb{R}^3$ with a covariance matrix as

$$G(\mathbf{x}) = e^{-\frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x}}. \tag{1}$$

And the covariance matrix Σ is defined as the multiplication of a rotation matrix $\mathbf{R}$ and a scaling matrix $\mathbf{S}$:

$$\Sigma = \mathbf{RSS}^T\mathbf{R}^T. \tag{2}$$

Each point also has rendering attributes, including opacity $\mathbf{o} \in \mathbb{R}$ and color in spherical harmonic (SH) form $\mathbf{c} \in \mathbb{R}^k$, where k is the number of spherical harmonic functions. In summary, each 3D Gaussian point has a set of attributes as $(\mathcal{X}, \mathbf{r}, \mathbf{s}, \mathbf{c}, \mathbf{o})$, and the collection of all Gaussians is denoted as $\mathcal{G}$.

The novel view image $\hat{I}$ with a viewing transform matrix $\mathbf{W}$ is rendered by splatting the 3D Gaussians $\mathcal{G}$. Furthermore, 3DGS introduces the differentiable view frustum and fast sorting techniques, which allow for extremely fast rendering. The differentiable splatting method is denoted as S, where $\hat{I} = S(\mathcal{G}, \mathbf{W})$.

B. 4D Gaussian Splatting

4D Gaussian Splatting (4DGS) [9] extends the applicability of 3DGS which is only applicable to static scenes, to dynamic scenes by introducing a Gaussian deformation field $\mathcal{F}$. It deforms the 3D Gaussians $\mathcal{G}$ at the canonical space to the arbitrary time space t, as $\mathcal{G}' = \mathcal{G} + \mathcal{F}(\mathcal{G}, t)$. The Gaussian deformation field $\mathcal{F}$ predicts the deformation of the pose $\mathcal{X}$, the rotation factor $\mathbf{r}$, and the scale factor $\mathbf{s}$ of each Gaussian, as $(\mathcal{X} + \Delta\mathcal{X}, \mathbf{r} + \Delta\mathbf{r}, \mathbf{s} + \Delta\mathbf{s})$. At time t, the rendered image using the viewing transform matrix $\mathbf{W}$ is denoted as $S(\mathcal{G}', \mathbf{W})$.

IV. Method

A. Articulated deformation field

We are considering dynamic scenes comprising articulated objects, where the scene contains up to K parts (though the actual number of parts may be fewer). A key characteristic

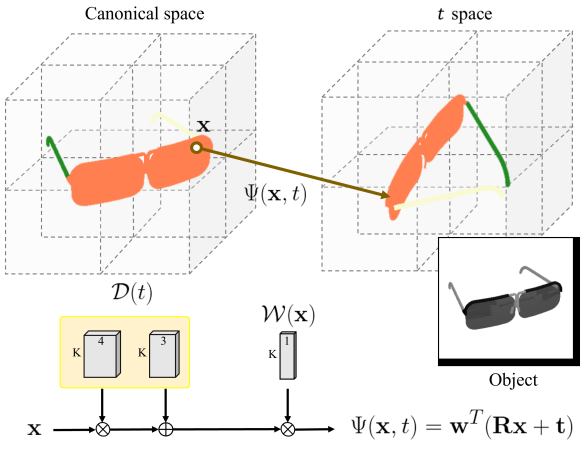

Fig. 2. Visualization of the structure of (a) a decomposition network and (b) a deformation network.

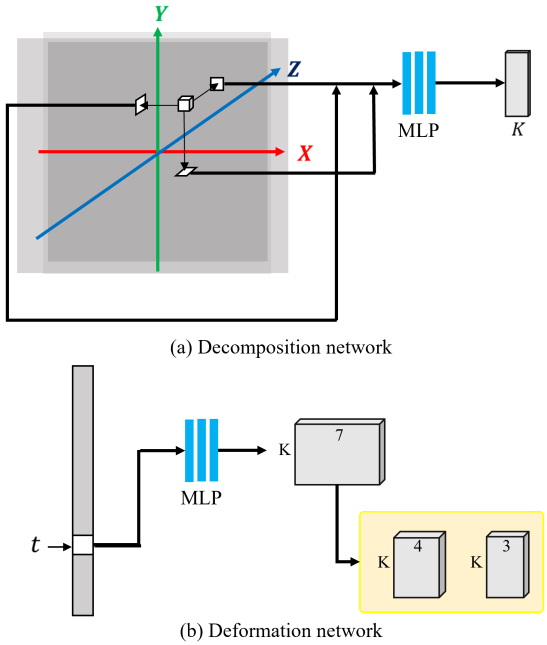

Fig. 3. Visualization of the articulated deformation field Ψ for the proposed model. The articulated deformation field $\Psi(\mathbf{x}, t)$ is formulated using the outcomes from $\mathcal{W}(\mathbf{x})$ and $\mathcal{D}(t)$.

of articulated objects is that spatial locations belonging to the same part in the canonical space can be mapped to arbitrary times t by the same 3D transformation. Therefore, the articulated deformation field can be defined as

$$\begin{aligned}\Psi(\mathbf{x}, t) &= \sum_{k=1}^{K} w_k(\mathbf{x})(\mathbf{R}_k(t)\mathbf{x} + \mathbf{t}_k(t)) \\ &= \mathbf{w}(\mathbf{x})^T(\mathbf{R}(t)\mathbf{x} + \mathbf{t}(t)),\end{aligned} \tag{3}$$

where $\mathbf{R}_i(t) \in \mathbb{R}^{3\times3}$ and $\mathbf{t}_i(t) \in \mathbb{R}^3$ represent the rotation and translation for the i-th part at time t, and $w_i(\mathbf{x}) \in \{0, 1\}$ represents whether the spatial location $\mathbf{x}$ belongs to the i-th part or not. $\mathbf{R}(t) \in \mathbb{R}^{K\times3\times3}$, $\mathbf{t}(t) \in \mathbb{R}^{K\times3}$, and $\mathbf{w}(\mathbf{x}) \in \mathbb{R}^K$ are aggregations of all rotations, translations, and probabilities. For the articulated deformation field, we introduce a decomposition network $\mathcal{W}(\mathbf{x}) : \mathbb{R}^3 \to \mathbb{R}^K$ and a deformation network $\mathcal{D}(t) : \mathbb{R} \to \mathbb{R}^{K\times7}$. The decomposition network utilizes a tri-plane structure, and the deformation network outputs a quaternion and translation, as illustrated in Figure 2. The overall articulated deformation field $\Psi(\mathbf{x}, t)$ is visualized in Figure 3.

B. The entire pipeline for training

First, we utilize 4DGS to generate a set of Gaussians $\{\mathcal{G}_1, \cdots, \mathcal{G}_F\}$ given a set of training cameras $\{(\mathbf{M}_1, t_1), \cdots, (\mathbf{M}_F, t_F)\}$, where $\mathbf{M}$ is the view matrix of the camera and t is the frame time, and F is the number of frames. For convenience, the Gaussian in the central frame and the time have been defined as the canonical Gaussian and canonical time as $(\mathcal{G}_{\text{can}}, t_{\text{can}}) = (\mathcal{G}_{F//2}, t_{F//2})$. We then create the articulated deformation field as described in Section IV-A and train it using the set of Gaussians. When training the articulated deformation field, the parameters of the generated Gaussians are frozen. As a result, we can obtain the decomposition of the canonical space and deformation from the canonical space to arbitrary times. The whole pipeline is visualized in Figure 1.

C. Loss function

This section explains the losses for training the articulated deformation field Ψ.

Photometric loss. Given a view matrix and the time $(\mathbf{M}, t)$, the photometric loss is the L1 color loss between the ground-truth image I and the rendered image with the deformed canonical Gaussians $\Psi(\mathcal{G}_{\text{can}})$ as $\hat{I} = S(\Psi(\mathcal{G}_{\text{can}}, t), \mathbf{M})$:

$$\mathcal{L}_{\text{photo}} = \|I - \hat{I}\|_1. \tag{4}$$

3D reconstruction loss. To efficiently train the articulated deformation model, we introduce a novel 3D reconstruction loss. The core idea of the 3D reconstruction loss is simple. First, we denote the point cloud of Gaussians $\mathcal{G}$ as $\mathcal{G}(\mathcal{X})$. When mapping the point cloud of the canonical space Gaussians $\mathcal{G}_{\text{can}}(\mathcal{X})$, to the arbitrary time t_i in the training camera using the articulated deformation field as $\Psi(\mathcal{G}_{\text{can}}(\mathcal{X}), t_i)$, it should closely resemble the $\mathcal{G}_i(\mathcal{X})$.

We propose a new method for measuring the distance between two point clouds, tailored for our articulated deformation field, due to unsatisfactory results observed with the chamfer distance commonly used for measuring distance between two point clouds. First, we voxelize the point clouds of each Gaussians into a voxel map of size $W \times H \times D$, denoted as $V(\mathcal{G}_i(\mathcal{X}))$, where W, H, and D represent the dimensions of the voxel map. We map $V(\mathcal{G}_{\text{can}}(\mathcal{X}))$ to time t_i as $\Psi(V(\mathcal{G}_{\text{can}}(\mathcal{X})), t_i)$, and define the 3D reconstruction loss as the L2 loss between $\Psi(V(\mathcal{G}_{\text{can}}(\mathcal{X})), t_i)$ and $V(\mathcal{G}_i(\mathcal{X}))$ as follows:

$$\mathcal{L}_{3D} = \frac{1}{F}\sum_{i=1}^{F} \|\Psi(V(\mathcal{G}_{\text{can}}(\mathcal{X})), t_i) - V(\mathcal{G}_i(\mathcal{X}))\|_2^2. \tag{5}$$

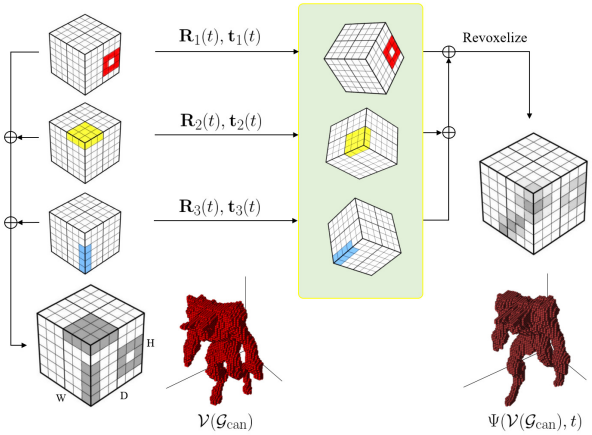

Fig. 4. Visualization of the process of obtaining $\Psi(V(\mathcal{G}_{\text{can}}(\mathcal{X})), t)$ from the articulated deformation field network. Each part is visualized with a different color - red, yellow, and blue - in the voxel regions, and the overall emptiness voxel map $V(\mathcal{G}_{\text{can}}(\mathcal{X}))$ is the sum of the emptiness voxel map for all parts. After applying the corresponding transformations to the emptiness voxel map for each part and revoxelizing, we obtain $\Psi(V(\mathcal{G}_{\text{can}}(\mathcal{X})), t)$.

To map $V(\mathcal{G}_{\text{can}}(\mathcal{X}))$ to time t_i, we transform $\mathcal{G}_{\text{can}}$ for all rotations and translations, then take the weighted sum according to the label vector, and finally voxelize them. A visual representation of the process of obtaining $\Psi(V(\mathcal{G}_{\text{can}}(\mathcal{X})), t)$ is given in Figure 4.

Total loss function. The total loss function is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{photo}} + \mathcal{L}_{\text{3D}} + \mathcal{L}_{\text{tv}}, \tag{6}$$

where $\mathcal{L}_{\text{tv}}$ is a grid-based total-variational loss [6], [5], [29], [9].

V. Experiments

We evaluate the performance on a 3D part segmentation task, conducting all experiments on a GeForce RTX 4070 Ti Super GPU with a maximum of 40 parts. The proposed method takes an average of 10 minutes on the D-NeRF dataset and 3 minutes on 4DGS alone, where the second stage averages 7 minutes.

A. 3D Part decomposition results

Datasets. To evaluate the performance of 3D part decomposition, monocular videos of deforming articulated objects with ground-truth decomposition label are necessary. We generate our own dataset using articulated objects from two datasets, the KinArt3D dataset [24] and the PartNet Mobility dataset [30], both of which provide kinematic models for articulated objects. We capture images and ground-truth labels of articulated objects with changing poses using the same camera views of the D-NeRF dataset [4].

Quantitative results. We conduct a quantitative performance comparison between the proposed method and the most relevant method, Watch It Move (WIM) [10]. Additionally, we use two studies, Segment Anything (SAM) [31] and UIS [32], both of which perform unsupervised part segmentation at the image-level without utilizing optical flow as input, as

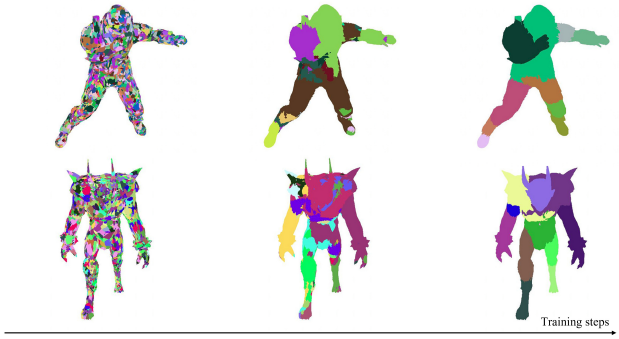

Fig. 5. Illustration of the optimization process. Starting from a random 3D part label, our method successfully estimates 3D parts in an unsupervised manner.

TABLE I
Quantitative results of 3D part decomposition on custom dataset.

Method	mIoU						
	Door	Drawer	Kettle	Lamp	Leg	Robot	Average
UIS [32]	0.81	0.53	**0.92**	0.67	0.55	0.81	0.72
SAM [31]	**0.94**	0.71	0.78	0.91	0.57	0.80	0.79
WIM [10]	0.60	0.67	0.88	0.79	0.86	0.85	0.78
Ours	0.92	**0.91**	**0.92**	**0.97**	**0.96**	**0.92**	**0.93**

baselines. The overall results are summarized in Table I. The results demonstrate that the proposed method outperforms the baselines. Image-level part segmentation algorithms [31], [32] decompose a higher number of parts than the actual motion-based parts because they do not consider the entire video sequence. This, in turn, contributes to a decrease in performance. WIM [10] also shows inferior performance compared to the proposed method. This is because it was originally designed to train on multi-view videos rather than monocular videos, suggesting a degradation in performance on monocular videos.

Qualitative results. For the quantitative measure, we utilize the mean Intersection over Union (mIoU) between the predicted labels and the ground-truth labels. The 3D part decomposition results, on the custom dataset, of the proposed methods and baselines are shown in Figure 6. We observe that the decomposition results obtained by the proposed method are close to the ground-truth and outperform those of other methods overall The decomposition results for the D-NeRF dataset are shown in Figure 7. The proposed method shows reasonable outcomes from a qualitative perspective. Figure 5 illustrates the progressive enhancements in the decomposition for the "jumping jack" and "standup" in the D-NeRF dataset. It depicts labels assigned to occupied voxels at the canonical space, demonstrating that even starting from completely random labels, meaningful partitions emerge based on motions. Notably, even without additional post-processing steps for part merging, redundant parts are eliminated.

Point cloud distance metric. As mentioned in Section IV-C, instead of the proposed 3D reconstruction loss, the chamfer distance can be used in the 3D reconstruction loss. However,

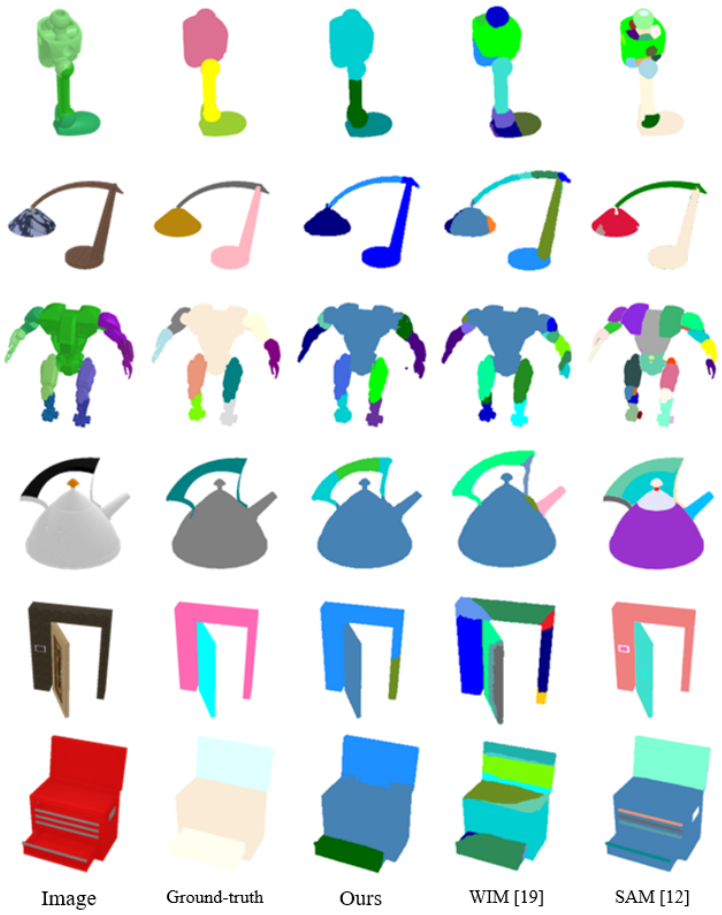

Fig. 6. Qualitative results on the custom dataset. We can see that the decomposition results of the proposed method are superior compared to the baselines.

Fig. 7. Qualitative results on the D-NeRF dataset. We can see that the proposed method successfully synthesizes novel-view images and predicts 3D part labels.

experimentally, when using the chamfer distance, we observe that although the decomposition results are seem to be reasonable, the deformation results are adversely affected. Figure 8 shows a common failure case when using the chamfer distance in the "Hellwarrior" scene, where two legs are switched.

VI. Limitations

The proposed method has several limitations. The main limitation is that the proposed method faces challenges in dividing overly small parts and exhibits instability in results at joint regions where two parts meet. Furthermore, the proposed articulated deformation field learns to assign points

Fig. 8. Failure case when using the chamfer distance loss.

to the same part based solely on their motion similarity. As a result, decomposition results often assign points to the same part even if they are spatially distant but have similar motion. This is not intuitive for human perception and needs improvement.

VII. Conclusion

In this paper, we introduce a novel method for inferring the motion-based parts of articulated objects from monocular videos leveraging Gaussian splatting. Departing from conventional approaches, our decomposition model offers flexibility by avoiding reliance on primitive geometries and does not require additional information such as optical flow during training. Inspired by ICP techniques, we tailor a transformation function to the motion patterns of articulated objects and address the challenges of the chamfer distance with voxelized representations of Gaussians. In experiments, the proposed method shows superior results in unsupervised 3D part decomposition. Future work can focus on overcoming the aforementioned limitations of the proposed method.

References

[1] L. Liu, W. Xu, H. Fu, S. Qian, Q. Yu, Y. Han, and C. Lu, "Akb-48: A real-world articulated object knowledge base," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 809–14 818.

[2] M. Mittal, D. Hoeller, F. Farshidian, M. Hutter, and A. Garg, "Articulated object interaction in unknown scenes with whole-body mobile manipulation," in *2022 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2022, pp. 1647–1654.

[3] G. Schiavi, P. Wulkop, G. Rizzi, L. Ott, R. Siegwart, and J. J. Chung, "Learning agent-aware affordances for closed-loop interaction with articulated objects," in *2023 IEEE International Conference on Robotics and Automation*. IEEE, 2023, pp. 5916–5922.

[4] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural radiance fields for dynamic scenes," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 318–10 327.

[5] J. Fang, T. Yi, X. Wang, L. Xie, X. Zhang, W. Liu, M. Nießner, and Q. Tian, "Fast dynamic radiance fields with time-aware neural voxels," in *SIGGRAPH Asia 2022 Conference Papers*, 2022, pp. 1–9.

[6] A. Cao and J. Johnson, "Hexplane: A fast representation for dynamic scenes," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 130–141.

[7] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–14, 2023.

[8] W. Yifan, F. Serena, S. Wu, C. Öztireli, and O. Sorkine-Hornung, "Differentiable surface splatting for point-based geometry processing," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–14, 2019.

[9] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, "4d gaussian splatting for real-time dynamic scene rendering," *arXiv preprint arXiv:2310.08528*, 2023.

[10] A. Noguchi, U. Iqbal, J. Tremblay, T. Harada, and O. Gallo, "Watch it move: Unsupervised discovery of 3d joints for re-posing of articulated objects," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3677–3687.

[11] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, 1992.

[12] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International journal of computer vision*, vol. 13, no. 2, pp. 119–152, 1994.

[13] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2d and 3d point matching: pose estimation and correspondence," *Pattern recognition*, vol. 31, no. 8, pp. 1019–1031, 1998.

[14] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.

[15] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *Proc. of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 605–613.

[16] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[17] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *arXiv preprint arXiv:1706.02413*, 2017.

[18] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proc. of the IEEE/CVF International Conference on Computer Vision*, 2019.

[19] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proc. of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 259–16 268.

[20] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[21] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, "Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[22] D. Tzionas and J. Gall, "Reconstructing articulated rigged models from rgb-d videos," in *European Conference on Computer Vision*, 2016.

[23] U. M. Nunes and Y. Demiris, "Online unsupervised learning of the 3d kinematic structure of arbitrary rigid bodies," in *Proc. of the IEEE/CVF International Conference on Computer Vision*, 2019.

[24] J. G. Choy, G. Cha, and S. Oh, "Unsupervised 3d link segmentation of articulated objects with a mixture of coherent point drift," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7124–7131, 2022.

[25] W.-C. Hung, V. Jampani, S. Liu, P. Molchanov, M.-H. Yang, and J. Kautz, "Scops: Self-supervised co-part segmentation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 869–878.

[26] A. Ziegler and Y. M. Asano, "Self-supervised learning of object parts for semantic segmentation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 502–14 511.

[27] S. Liu, L. Zhang, X. Yang, H. Su, and J. Zhu, "Unsupervised part segmentation through disentangling appearance and shape," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8355–8364.

[28] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in *Proc. of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5865–5874.

[29] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa, "K-planes: Explicit radiance fields in space, time, and appearance," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 479–12 488.

[30] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang *et al.*, "Sapien: A simulated part-based interactive environment," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 097–11 107.

[31] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[32] W. Kim, A. Kanezaki, and M. Tanaka, "Unsupervised learning of image segmentation based on differentiable feature clustering," *IEEE Transactions on Image Processing*, vol. 29, pp. 8055–8068, 2020.