IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Interactive Robotic Moving Cable Segmentation by Motion Correlation

Ondřej Holešovský[1], Radoslav Škoviera[2], and Václav Hlaváč[2]

***Abstract*—Manipulating tangled hoses, cables, or ropes can be challenging for both robots and humans. Humans often approach these perceptually demanding tasks by pushing or pulling tangled cables and observing the resulting motions. We follow a similar idea to aid robotic cable manipulation. In this letter, we integrate visual and proprioceptive perception to segment a grasped cable by moving it even when the robot or the grasped cable sometimes perturb neighboring cables. We formulate the cable interactive segmentation problem in such a way that our methods do not require robot arm segmentation masks. Furthermore, a novel grasp sampling method can propose new cable grasp points given a partial cable segmentation to improve the segmentation via additional cable-robot interaction. We evaluate the proposed *motion correlation* (MCor) method on data sequences recorded by our physical robotic setup and show that the method outperforms an earlier *motion segmentation* (MSeg) baseline.**

***Index Terms*—Object detection, segmentation and categorization, perception for grasping and manipulation, data sets for robotic vision, cable motion, optical flow.**

I. Introduction

MANIPULATING one-dimensional deformable objects such as cables, hoses or ropes (henceforth referred to as "cables" for brevity), especially when cluttered, is challenging both for humans and robots. It is due to self-occlusions, high-dimensional state space, uniform visual appearance, and complex interaction dynamics. Imagine, for example, that a robot should replace a specific damaged hose in the scene shown in Fig. 1. There are passive computer vision methods [1], [2], [3] for segmenting individual cable instances. However, these methods struggle with occlusions or complex intersections of multiple cables [4]. Novel methods are therefore needed.

We are inspired by the way humans interactively discover the topology of cluttered cables. When humans find it too hard to visually infer whether two cable segments are directly linked, they grasp and pull or push one of them. The motion visually distinguishes the grasped cable from the clutter. [4] proposed a dataset and method for segmenting individual moving cables when only one cable in the scene is moving. In this work, we integrate visual and proprioceptive perception to aid robotic cable exploration even when the robot or the grasped cable sometimes perturb neighboring cables.

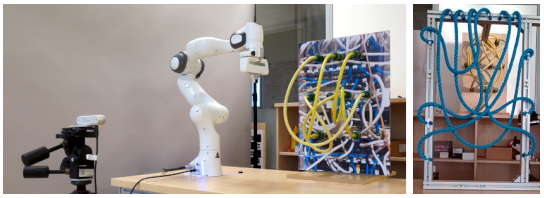

Fig. 1: The robotic setup. An RGB-D camera (left), a robotic arm (middle), a task board with tangled hoses or a frame with ropes (right).

As our robots are too large to manipulate thin cables gently, we recorded image sequences featuring ropes or garden hoses being manipulated by a robotic arm. Fig. 1 shows the robotic setup we used in our experiments.

The contributions of this letter include:

1) A *motion correlation* (MCor) method able to segment a grasped cable by moving it in a cluttered environment even when the cable or the robot sometimes perturb neighboring cables.
2) A grasp sampling method which can propose new cable grasp points given a partial cable segmentation to improve the cable segmentation via additional cable-robot interaction.
3) A formulation of the cable motion segmentation problem which does not require robot arm segmentation masks.
4) An evaluation of the proposed methods on data sequences newly recorded by our physical robotic setup.

Section II discusses the related work. We introduce the MCor and grasp sampling methods as well as the *motion segmentation* (MSeg) baseline in Section III. Section IV presents the implementation in a physical robotic setup, Section V describes the experiments and their results. Section VI discusses the results and concludes the letter.

II. Related Work

Existing interactive segmentation approaches have been tested mostly on rigid objects. Several methods presented in

Received 20 January 2025; accepted 2 May 2025. Date of publication 29 May 2025. This letter was recommended for publication by Associate Editor J. Zhu and Editor M. Vincze upon evaluation of the reviewers' comments. This work was supported in part by the European Union under the project Robotics and Advanced Industrial Production (reg. no. CZ.02.01.01/00/22_008/0004590) and in part by projects TAČR no. FW08010076 and GAČR no. 23-04080L. (Corresponding author: Ondřej Holešovský.)

[1]Ondřej Holešovský is with the Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, Jugoslávských partyzánů 1580/3, 160 00 Prague, Czech Republic and with Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 166 27 Prague, Czech Republic (e-mail: research@holesovsky.me)

[2]Radoslav Škoviera and Václav Hlaváč are with the Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, Jugoslávských partyzánů 1580/3, 160 00 Prague, Czech Republic (e-mail: radoslav.skoviera@cvut.cz; vaclav.hlavac@cvut.cz)

Code, dataset, and videos are available at https://github.com/holesond/cmcor and https://doi.org/10.5281/zenodo.15356102.

Digital Object Identifier (DOI): 10.1109/LRA.2025.3574960

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

the literature can segment cables passively from images [5], [6], [3], [2], [1].

A. Cable perception

Cable segmentation is challenging because cables are often of uniform appearance without distinctive features. Several cable detection or segmentation methods in the literature simplified the task by having a single cable in the scene [7], [8] or by using color thresholding to segment the cables from the background [7], [9], [10], [11], [12], [13], [14].

State-of-the-art passive cable instance segmentation methods Ariadne+ [6], FASTDLO [3], RT-DLO [2], mBEST [1] often start with segmenting all the cables from the background in an image using a semantic segmentation neural network such as DeepLabV3+ [5]. They find individual cable instances in the foreground segmentation masks by processing skeletonized cable segments or graph-based representations. [3], [2], [1] did not report results for scenes with multiple overlapping cables and severe occlusions and such scenes were outside the scope of mBEST [1]. DLO3DS [15] estimated the 3D shape of static cables from multiple views captured by an eye-in-hand 2D camera. It relied on cable instance segmentation provided by FASTDLO. In [16], a combination of the Segment Anything large vision model with a post-processing method outperformed [5], [3] in segmenting a cable from the background. More recently, a Perceiver-inspired architecture segmented cable instances in a color image given a specific text-based prompt [17].

Classifiers or keypoint detectors can replace cable state estimation algorithms when task-specific human-labeled training data is available [12], [18], [13], [19], [20]. They can propose interaction keypoints, detect endpoints, classify knots, or refine grasps. Nevertheless, most of these approaches assumed that the cables were segmentable from the background by color thresholding.

[21], [22], [23] tracked a cable across multiple video frames given its segmentation in the first frame.

Our work exploits the motion of a cable of interest to simplify the cable perception task, even in complex scenes with multiple overlapping cables and severe occlusions.

B. Interactive segmentation

Interactive perception is the exploitation of forceful robot-environment interactions to simplify and enhance perception [24], [25]. Interactive segmentation [26], a more specific category of interactive perception methods, interacts with the environment and segments it into a set of movable objects based on the observed motion. It is computationally efficient and requires little prior knowledge about the environment.

Interactive segmentation usually needs a visual motion signal to segment the moving objects. We use dense optical flow for interactive segmentation [27], [28], [29], [30]. Compared to optical flow, intensity change detection [26] performs poorly when the moved object and the background are of similar color [28] or when multiple objects move [30]. We believe that we cannot apply sparse feature tracking [29], [31] or rigid object trackers [31] to most cables due to their uniform visual appearance and deformability.

There are several ways to segment the objects from the motion signals. Optical flow clustering [27], [30] followed by the rejection of rotating objects and the segments not overlapping with the initial gripper location [30] relied on the rigid object assumption. Optical flow thresholding can segment a rigid or deformable moving object when nothing else moves in the scene [28], [4], e.g. when a passive segmentation method segments the robot arm. Optical flow or object trackers can also track segmentation hypotheses between image frames [29], [31], [32], [33], [34]. The hypothesis representations used for that are diverse. [29] proposed a probabilistic segmentation framework to update an octree neighborhood graph, where each node represented a voxel and each edge encoded their similarity. [31] sampled segmentations from a segmentation tree generated by Convolutional Oriented Boundaries. [32] generated segmentation hypotheses with confidence estimates by prompting the Segment Anything Model (SAM). [33] tracked undersegmented passive instance segmentation masks of rigid objects across multiple pushing interactions to obtain more reliable segmentations. [34] performed interactive rigid object segmentation while assuming that several points could be tracked on each object and that an initial (under)segmentation was available.

[4] contributed a new moving cable dataset automatically annotated with optical flow and instance segmentation masks. They also proposed several cable motion segmentation methods which assumed that only a single cable was moving in the scene and that the segmentation masks of the arm moving the cables were available.

The MCor method presented in this work only slightly moves the grasped cable in different directions. It segments the cable by correlating the robot gripper motion with the cable motion measured by an optical flow estimator. MCor uses multiple robot actions to segment the cable of interest in the last frame of a recorded image sequence. Unlike [4], it can segment the grasped cable from neighboring cables moving due to perturbations. We also reformulate the interactive cable segmentation problem to avoid the need for the arm segmentation masks. We are not aware of any other method addressing the same problem of interactive cable segmentation.

III. Methods

The proposed method relies on the robot exploring a cable by grasping it at one or more locations and performing multiple actions at each location. Each action consists of a symmetric linear movement, e.g., along to the cable axis.

The method records a data sequence of color images and gripper positions. It divides the data sequence by action, such that the images $\{I_{a,k}\}$ and the gripper positions $\{g_{a,k}\}$ are indexed by the action index $a \in \{1, ..., A\}$ and the within-action sample index $k \in \{1, ..., N_a\}$, where A is the total number of actions in the data sequence and N_a is the number of data samples of action a. To ensure spatially uniform sampling, we subsample the images and gripper positions such that $||g_{a,k} - g_{a,k+1}||_2 \geq \delta_g$, where δ_g is the minimum gripper sample displacement.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

To segment moving cables, the MCor and MSeg methods compute optical flow vectors $\phi_{a,k}(p) \in \mathbb{R}^2$ for all discrete image pixel locations $p \in \mathbb{R}^2$ between each image $I_{a,k}$ and the target image I_{A,N_A}, which is the last image of the last action of the recorded data sequence. The motion segmentation/correlation methods express all the flow vectors in the target image frame, such that ideally $I_{a,k}(p+\phi_{a,k}(p)) \approx I_{A,N_A}(p)$ holds for all non-occluded pixels p, where $I_{a,k}(p)$ is the image color or brightness sampled at pixel p. The gripper/arm is outside of the field of view in the target frame, which ensures that the arm does not cause any optical flow. If the optical flow estimator incorrectly matches the missing arm to something else, MCor can still largely suppress the wrong flow as it is usually not very correlated to the gripper motion.

An action with index a is a robot periodically moving a cable initially grasped at position $g_{a,1}$ between the positions $(g_{a,1} + \Delta_a)$, $g_{a,1}$, $(g_{a,1} - \Delta_a)$, and back to $g_{a,1}$. Δ_a is a constant action-specific vector that defines the direction and amplitude of the movement, 1 is the index of the first (reference) image and of the first gripper position of action a. The robot repeats the periodic motion K_p times in each action. We let the robot perform two actions for each initial grasp point $g_{a,1}$. One action moves the gripper along the cable axis projected to the camera image plane, the other perpendicularly to the cable axis. We note, however, that the robot could easily execute a more diverse set of actions specified by different action vectors Δ_a.

MCor and MSeg output an action vote image $V(p) = \sum_{a=1}^{A} M_a(p)$ for an explored cable, where $M_a(p)$ is a motion mask from action a. $M_a(p) = 1$ only at those pixels p reported as moving during action a and $M_a(p) = 0$ otherwise. We can say that action a labels as moving or votes for only those pixels p where $M_a(p) = 1$.

The segmentation methods assume that an initial cable segment to grasp is given for each cable instance which should be explored and segmented. In our experiments with the physical robot, a human provided the initial grasp point by drawing the short cable segment to grasp into the camera image frame on a computer screen. Alternatively, passive semantic cable segmentation methods or known cable endpoint locations (connectors, sockets) could provide the initial grasp.

To enable the robot to grasp a selected cable segment, the methods need to estimate the 3D cable center and axis given the 2D cable segment in an RGB-D image. The methods compute the segment center as the geometric median of the cable segment 3D points. Assuming that the segment is longer than it is wide, the cable axis is the principal axis computed by PCA from the set of cable segment 3D points within an inlier distance from the segment center.

Once an interactive cable segmentation method (partially) segments the initially grasped cable by moving it, a grasp sampling algorithm can propose the next suitable cable segment to grasp and move to increase the recall while preserving precision of the overall interactive segmentation process. We describe our grasp sampling algorithm in Section III-C.

We describe two interactive cable segmentation methods in this section. One is the MSeg baseline, an adaptation of MfnProb FT from [4]. The other is the novel MCor method we propose.

A. Motion Segmentation (MSeg) Baseline Method

The MSeg baseline method is MfnProb FT moving cable segmentation method from [4]. The baseline assumes that there is only one cable moving in the scene and that any perturbations of neighboring cables are negligibly small. It estimates moving cable segmentations in each image of a sequence. We adapted it in this work, such that it segments cables only in the target frame based on the motion detected at image $I_{a,k}$ using the optical flow magnitude

$$||\phi_{a,k}(p) - \phi_{a,1}(p)||_2 > \tau_m, \tag{1}$$

where τ_m is a motion threshold. Each pixel p where the flow magnitude is larger than τ_m adds a one into the segmentation count image $S_a(p)$. The algorithm sets $M_a(p) = [\![S_a(p) > \tau_s N_a]\!]$, where τ_s is a relative segmentation count threshold ($0 \leq \tau_s \leq 1$) and N_a is the total number of input images of action a.

B. Motion Correlation (MCor)

To segment a grasped cable from other cables, we propose the MCor algorithm that leverages the predictable movement of the grasped cable when manipulated in multiple directions. Neighboring cables can be perturbed only if they are in contact with the grasped cable or the robot. For loosely entangled cables, repeated gripper motions in different directions typically move the grasped cable predictably while occasionally disturbing the neighboring cables. Our proposed algorithm exploits this predictability to effectively segment the grasped moving cable from its surroundings. See Fig. 2 for a visual summary of the method.

For each sample (a, k), the algorithm computes the gripper displacement along the motion direction,

$$d_{g,a,k} = (g_{a,k} - g_{a,1}) \cdot \frac{\Delta_a}{||\Delta_a||_2}. \tag{2}$$

In each image $I_{a,k}$, the algorithm processes only the pixels p where $||\phi_{a,k}(p) - \phi_{a,1}(p)||_2 > \tau_n$, where τ_n is a noise suppression threshold. For each such pixel, it computes the optical flow displacement in meters along the principle flow direction $\psi_a(p)$ (unit vector) during action a as follows

$$d_{\phi,a,k}(p) = \frac{Z_{a,k}(p)}{f}(\phi_{a,k}(p) - \phi_{a,1}(p)) \cdot \psi_a(p), \tag{3}$$

where f is the camera focal length in pixels and $Z_{a,k}(p)$ is the depth of pixel p in image $I_{a,k}$ in meters. The equation assumes that the image has already been undistorted and that the camera pixel shape is (almost) square. (One could generalize to rectangular pixels by scaling the horizontal flow coordinates by f_x and the vertical ones by f_y to compute the 2D metric flow.)

The method estimates $\psi_a(p)$ from the relative flow vectors $(\phi_{a,k}(p) - \phi_{a,1}(p))$ using online PCA with online flow covariance computation.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

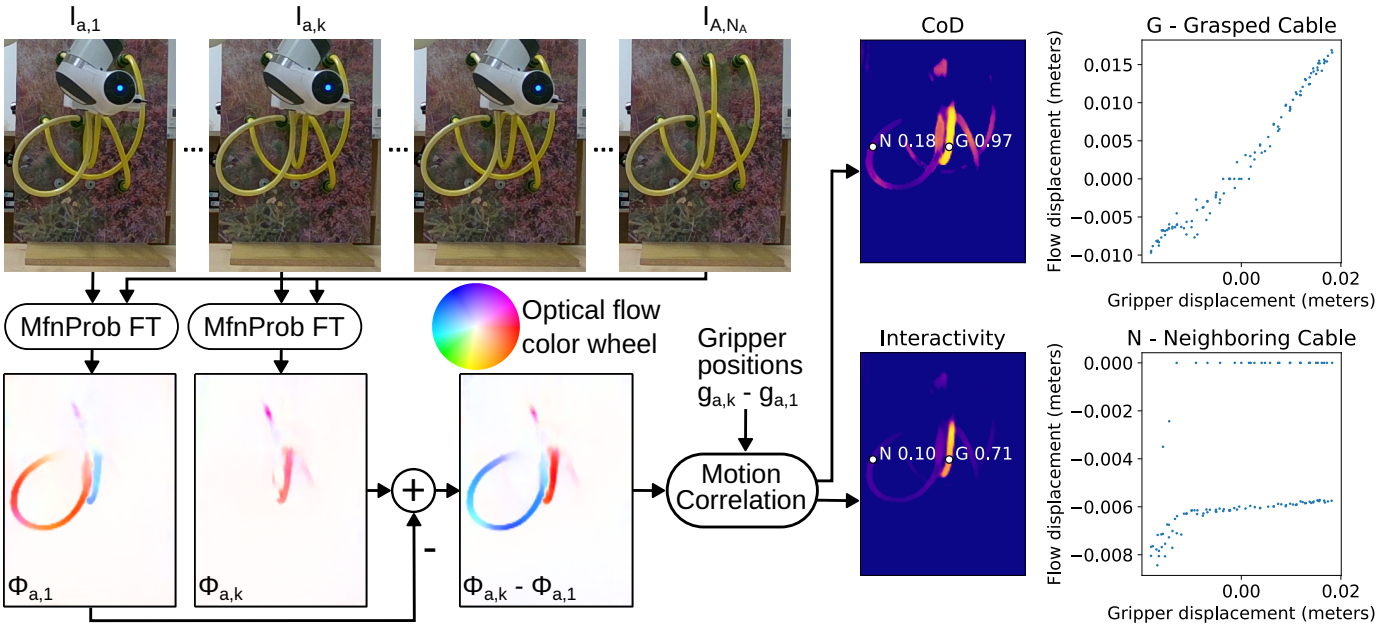

Fig. 2: MCor method data flow. Method components are in round boxes. MfnProb FT [4] estimates the optical flow.

Fitting a linear model h_a to predict flow displacement $d_{\phi,a,k}(p)$ given gripper displacement $d_{g,a,k}$,

$$d_{\phi,a,k}(p) \approx h_a(p, d_{g,a,k}) = c_{a,0}(p) d_{g,a,k} + c_{a,1}(p), \quad (4)$$

and computing the coefficient of determination (CoD) $R^2_a(p)$ is the last step of the algorithm. The algorithm computes the linear model parameters $c_{a,0}(p)$, $c_{a,1}(p)$ and $R^2_a(p)$ online from the first- and second-order moments of $d_{\phi,a,k}(p)$ and $d_{g,a,k}$ to reduce its memory requirements. Pixels p where $|c_{a,0}(p)| > c_{0,min}$ (interactivity threshold) and $R^2_a(p) > R^2_{min}$ (CoD threshold) are labeled as moving, i.e. $M_a(p) = 1$, other pixels p as static, i.e. $M_a(p) = 0$. The first condition ensures that only sufficiently interacting cable segments are labeled as moving. The second condition suppresses the areas where the relationship between the gripper and flow displacements significantly differs from a linear function.

C. Grasp Sampling

We further propose a grasp sampling algorithm, see Algorithm 1, to enhance the recall while preserving the precision of the overall interactive segmentation process by identifying the next suitable cable segment to grasp and move, given a partial cable segmentation (voteImage) and previous grasp points (previousGrasps).

The algorithm first prefers segments voted for by the highest number of different actions to reduce the possibility of grasping a wrong cable. Exploration efficiency is its second priority as it returns segments by their distance from previous grasps. The minimum distance of the sampled grasp from its nearest previous grasp is D_{min}.

The function graspFromPixel attempts to construct a valid grasp meeting several requirements:

1) A sampled grasp segment has a minimum and maximum acceptable size (image area) and a minimum acceptable eccentricity. This reduces the possibility of sampling too small noise segments or segments which are not elongated enough and thus do not resemble cables. The maximum area threshold ensures sufficiently dense sampling of the segmentation image.
2) The estimation of the 3D cable segment axis is possible using the sampled 2D segment and the depth image, i.e. there is a sufficient number of depth measurements available in the sampled cable segment area.
3) The angle between the cable segment axis and the image plane is sufficiently low. Pose and depth estimation of cable segment axes which are close to normal to the image plane is less accurate due to a lower number of pixel samples per a metric unit of cable length.

Finally, a motion planner tries to plan a robot end effector trajectory to grasp each sampled segment. If the planning fails, e.g. due to pose unreachability or robot (self) collisions, the grasp sampler samples another grasp.

Algorithm 1 Grasp sampling algorithm.

```
function GRASPSAMPLER(voteImage, previousGrasps)
    ν_max ← max(voteImage)
    blacklist ← {∅}
    for each ν ∈ [ν_max, ν_max − 1, ..., 1] do
        mask ← thresholdImage(voteImage, ν)
        mask ← removeBlacklisted(mask, blacklist)
        sortedPixels = pixelsByGraspDistance(
            mask, previousGrasps, D_min)
        for each p ∈ sortedPixels do
            grasp ← graspFromPixel(p, mask)
            if grasp is valid then
                blacklistGrasp(
                    grasp, sortedPixels, blacklist)
                yield grasp
```

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

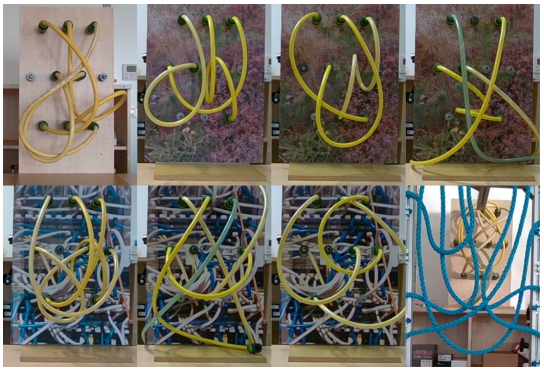

Fig. 3: The eight cable configurations of the dataset.

TABLE I: The main features of the dataset.

Property	Count	Comment
Cable configurations	8	
Cables explored	27	21 hoses, 6 ropes
Backgrounds	4	plain wood, bushes, cables, hoses
Validation sequences	8	2 977 images
Test sequences	66	26 305 images

TABLE II: Parameter values used in our experiments.

Parameter name	Symbol	Value
Min. gripper sample displacement	δ_g	0.001 meters (0.5 px)
Action amplitude	$\|\|\Delta_a\|\|_2$	0.02 meters (10 px)
Number of motion periods	K_p	3
Motion threshold (only for MSeg)	τ_m	2.0 pixels (from [4])
Noise suppression threshold	τ_n	0.67 pixels
Relative seg. count threshold	τ_s	see Table III
Interactivity threshold	$c_{0,min}$	see Table III
Coefficient of determination thr.	R^2_{min}	see Table III
Min. exploration distance	D_{min}	0.1 meters

IV. Implementation

We performed real-world cable manipulation experiments using a Franka Emika Panda robot controlled via MoveIt2 in ROS2. An Intel RealSense D456 RGB-D camera watched the robot and the task board mounted next to it, see Fig. 1. Color and depth image resolution was 1280×720 pixels at 15 FPS framerate. 15.4 mm thick hoses or 14 mm thick ropes were at the depth of ca. one meter. As the cable depth variability relative to the mean cable depth was low in our experiments, we used the gripper depth in the camera coordinate frame in place of $Z_{a,k}(p)$ for simplicity. We mounted the ropes to a rectangular aluminum frame. We attached the hoses to the hose task board using garden hose connectors. To obtain variously complex scene backgrounds, we mounted replaceable cardboard sheets behind the garden hose connectors. We printed a different background image on each sheet.

We used M3T multi-body tracker [35] for camera-robot calibration. A constant transformation matrix or a human selecting four keypoints on the robot in a color image approximately initialized the tracker when the robot was in the "ready" pose. M3T then iteratively refined the camera-robot pose estimate by tracking the robot arm in the RGB-D images.

MfnProb FT deep neural network model trained by [4] estimated the optical flow for MCor and MSeg. It processed color images cropped to the task board area. We scaled up the images $1.5\times$ because MfnProb tends to ignore the motion of cables thinner than ca. ten pixels. To compute the optical flow in parallel with the motion correlation algorithm, we implemented the computation in a two-stage pipeline. The first stage computed the optical flow on a GPU (NVIDIA GeForce RTX 2080 Ti), the second stage ran the motion correlation implemented in Numpy on a CPU (Intel Core i9-9900K CPU (3.60GHz)).

MoveIt2 built an octomap from scene point clouds to avoid collisions when reaching for the grasps.

V. Experiments

We evaluated the cable segmentation methods on a dataset of 66 sequences recorded with our robotic setup. We manually annotated the grasped cable ground truth segmentation in the target image of each dataset sequence. Each sequence consists of two actions performed at one grasp point. Table I summarizes the main features of the recorded dataset and Fig. 3 shows its eight cable configurations.

High cable segmentation precision with low recall is usually more valuable in a robotic cable exploration scenario than high recall with low precision [4]. Precision typically determines manipulation success, e.g. grasping the correct cable, whereas recall affects task efficiency, e.g. the number of actions required to explore the cable. Therefore we prefer the F_β score over the intersection over union (IoU) as the main measure of cable segmentation performance. We use F_β with $\beta < 1$ to increase the importance of precision over recall. The lower the β, the higher is the weight of precision in the F_β computation. $F_\beta = (\beta^2 + 1)/(\beta^2\mathcal{R}^{-1} + \mathcal{P}^{-1})$, where $\mathcal{R}$ is recall and $\mathcal{P}$ is precision.

Table II lists the parameter values we used. δ_g is the gripper displacement causing approx. a 0.5 pixel optical flow magnitude. $||\Delta_a||_2$ is the displacement causing approx. 10 pixel optical flow magnitude. It is the practical minimum for MCor to segment cables with up to $5\times$ motion signal attentuation (see $c_{0,min}$ in Table III and τ_m (for MSeg) in Table II). Larger displacements could also work but they will more likely cause collisions with other cables or rigid obstacles or pull the grasped cable more than its slack allows. $K_p = 3$ is roughly the minimum number of motion periods required to gather sufficient data for MCor. We set τ_n to the 0.9999 quantile of the optical flow magnitude predicted by MfnProb FT for a static scene. We chose the threshold values ($c_{0,min}$, R^2_{min}, τ_s) to maximize the F_β score on the validation set for several different β values, see Table III. The predicted segmentation of each sequence was the union of the segmentations from both actions at a single grasp point. All three thresholds tend to grow with decreasing β to increase the segmentation precision.

A. Segmentation from a single grasp

Table IV presents the test set evaluation results for each β when the methods output for each sequence the union ($V(p) \geq 1$) or the intersection ($V(p) = 2$) of the segmentations predicted for both actions at a single grasp. In both

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE III: Parameters optimizing F_β score on the validation set for different β values.

β	τ_s	$c_{0,min}$	R^2_{min}
1.0	0.31	0.17	0.66
0.5	0.47	0.20	0.77
0.4	0.47	0.18	0.86
0.3	0.47	0.36	0.92

TABLE IV: Mean test metrics of methods computing the union ($\cup$) or the intersection ($\cap$) of segmentations from two actions of each sequence. The best results are in bold.

Method	β	Recall↑	Precision↑	F_β ↑	IoU↑
MSeg $\cup$	1.0	0.6831	0.4296	0.4796	0.3369
No gripper ablation $\cup$	1.0	**0.7224**	0.4033	0.4697	0.3284
MCor (ours) $\cup$	1.0	0.6197	**0.5472**	**0.5521**	**0.4031**
MSeg $\cup$	0.5	0.5825	0.4803	0.4702	0.3323
No gripper ablation $\cup$	0.5	**0.6749**	0.4310	0.4383	0.3282
MCor (ours) $\cup$	0.5	0.5446	**0.6615**	**0.6127**	**0.4218**
MSeg $\cup$	0.4	0.5825	0.4803	0.4721	0.3323
No gripper ablation $\cup$	0.4	**0.6598**	0.4372	0.4388	0.3266
MCor (ours) $\cup$	0.4	0.4941	**0.7743**	**0.6888**	**0.4292**
MSeg $\cup$	0.3	0.5825	0.4803	0.4747	0.3323
No gripper ablation $\cup$	0.3	**0.6515**	0.4399	0.4395	0.3252
MCor (ours) $\cup$	0.3	0.3577	**0.9002**	**0.7584**	**0.3441**
MSeg $\cap$	0.4	0.2176	0.4291	0.2680	0.1275
No gripper ablation $\cap$	0.4	**0.2634**	0.4016	0.2829	0.1455
MCor (ours) $\cap$	0.4	0.1613	**0.9412**	**0.4787**	**0.1592**
MSeg $\cap$	0.3	0.2176	0.4291	0.2913	0.1275
No gripper ablation $\cap$	0.3	**0.2543**	0.3990	0.2961	**0.1408**
MCor (ours) $\cap$	0.3	0.0533	**0.9751**	**0.2966**	0.0533

cases, MCor outperforms the MSeg baseline in terms of F_β and precision. On the other hand, the baseline achieves higher recall. (We ommited the intersection results for β 1.0 and 0.5 as they are not Pareto-efficient.)

The "No gripper ablation" method in Table IV is an ablation of the MCor method. It segments moving cables by thresholding the standard deviation of the relative optical flow displacement $d_{\phi,a,k}(p)$ while ignoring the gripper motion information. Its precision and F_β scores are worse than the ones of MCor, suggesting that the gripper motion information facilitates precise moving cable segmentation.

Fig. 4 shows sample segmentations. MSeg and MCor used the parameters optimizing $F_{0.4}$ on the validation set. The MSeg baseline tends to segment multiple moving cables as the grasped cable (Fig. 4 b, c). It also sometimes wrongly segments the robot arm, which is present in most sequence images but absent in the target image (Fig. 4 a, b). In contrast, MCor produces cable segments mostly correctly marking the grasped cable, especially when considering the intersection of the segmentations from both actions (Fig. 4 b, two votes). Both MCor and MSeg can segment cables partially extending beyond the image boundary (Fig. 4 a).

Table V compares the test set evaluation results of MCor and of several state-of-the-art passive segmentation methods. We used the 2D grasp point in each target image to prompt SAM 2 or to retrieve the cable instance segmentation mask closest to the grasp point in the case of FASTDLO, mBEST, RT-DLO.

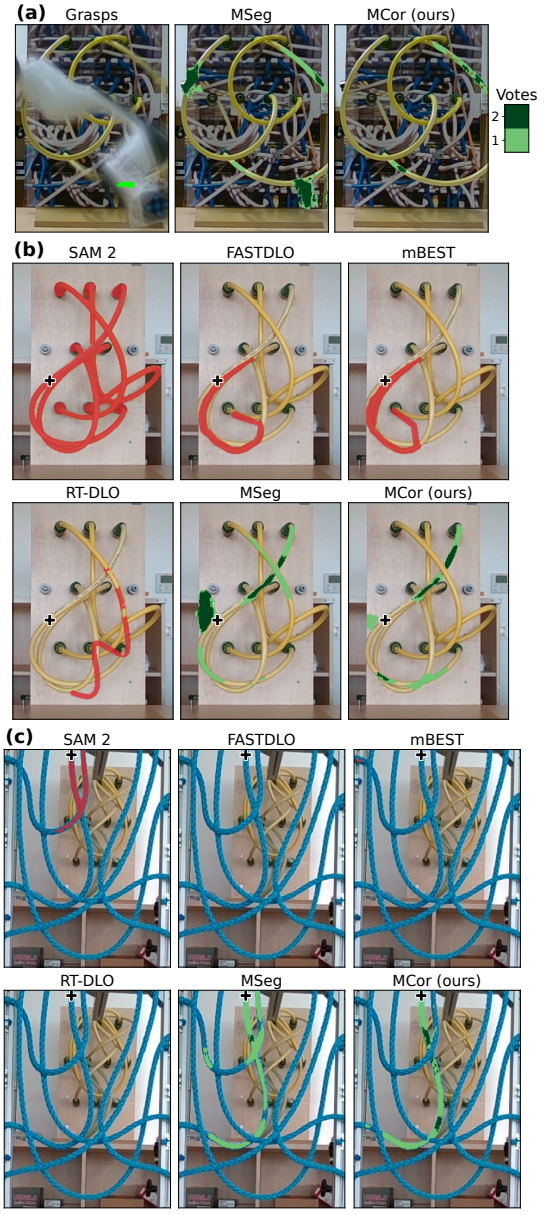

Fig. 4: Sample segmentations given a grasp/query point (Grasps image or "+" marker). The shades of green in the MSeg and MCor plots indicate the values of $V(p)$, i.e. the number of gripper actions voting for a given pixel.

MCor (optimizing $F_{0.4}$) outperforms the passive segmentation methods in terms of $F_{0.4}$, precision, and IoU. SAM 2 reaches the highest recall as it usually segments multiple or all the cables visible in the image instead of only the grasped cable (Fig. 4). FASTDLO, mBEST and RT-DLO do not segment the blue ropes as their cable semantic segmentation networks were not trained on ropes. MCor and Mseg can segment only those cable pixels which are visible both in the target image and in most of the sequence images. Therefore the presence of the arm in most images reduces the maximum achievable recall of MCor and MSeg and handicaps them in the comparison with passive segmentation methods. We evaluated all the methods only on the arm-free target images.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE V: Mean evaluation metrics on the test set of MCor and passive segmentation methods. MCor either computes the union (∪) or the intersection (∩) of segmentations from two actions of each sequence. The best results are in bold.

Method	Recall↑	Precision↑	$F_{0.4}$ ↑	IoU↑
SAM 2 Image (L) [36]	**0.7370**	0.5222	0.4986	0.3599
FASTDLO [3]	0.2767	0.6924	0.3685	0.2236
mBEST [1]	0.2722	0.4711	0.3496	0.2097
RT-DLO [2]	0.1800	0.4275	0.1916	0.1305
MCor (ours) ∪	0.4941	0.7743	**0.6888**	**0.4292**
MCor (ours) ∩	0.1613	**0.9412**	0.4787	0.1592

TABLE VI: Mean method runtimes.

Method	Runtime (seconds per image)↓
MSeg baseline	0.047
MCor (ours) - two parallel stages	0.047
Stage 1: Optical flow (GPU)	0.043
Stage 2: Motion correlation (CPU)	0.039

We report the mean runtime per processed image for each method and for each stage of the two-stage MCor pipeline in Table VI. Both methods are similarly fast and are suitable for real-time inference. We note that they currently require the last (target) image of a sequence before they can start computing the optical flow. Although one could easily compute the flow using the initial image instead of the target image, the robot needs to have the segmentations computed in the target frame for subsequent grasp selection. Despite running on the GPU, the optical flow stage is slower than the motion correlation stage utilizing just a single CPU core.

B. Segmentation from multiple grasps

Table VII reports the success rates of the grasps selected from MCor predictions by our grasp sampling algorithm. Most of the proposed grasps were on the correct cable. The robot could reach the grasps more often on the ropes than on the hoses, as the stiffer hoses spanned a larger depth range and created more obstacles for motion planning with the wide gripper of the Franka robot. Grasps could be unreachable due to being outside of the robot workspace, due to obstacles in the octomap or failed motion planning.

Table VIII presents MCor evaluation results on sixteen cables when segmenting a cable given the first and all grasp points. We provided the first grasp point manually. The grasp sampling method automatically proposed the other grasp points for each cable. In this experiment, a positive segmentation from each action contributed a unit vote to a multi-grasp vote image. The MCor method used the parameters optimizing $F_{0.4}$ on the validation set. Table VIII reports the segmentation accuracy computed at several action thresholds, i.e. the numbers of actions voting for each segmented pixel. All grasps yield higher segmentation recall than a single grasp. At the two-action threshold, the segmentation from two grasps has slightly lower precision than the single grasp segmentation but its $F_{0.4}$ score is higher. Precision increases with more actions voting for a segmented cable pixel. Fig. 5 demonstrates how mean segmentation performance increases with more interactions and grasp points. Fig. 6 shows sample cable segmentation given one and five grasp points. The red curve is a spline fitted to the segments with ≥ 2 votes. Such splines correctly matched the endpoints of 8 out of 9 cables with at least three robot grasps.

TABLE VII: Automatically proposed grasps.

Grasp proposal category	Hoses Count	Hoses %	Ropes Count	Ropes %
Total grasp proposals	54	100	31	100
Correct & reachable	20	37	22	71
Correct & unreachable	32	59	9	29
Wrong, another cable	2	4	0	0
Wrong, background	0	0	0	0

TABLE VIII: MCor multi-grasp results. The better results for each action threshold (action thr.) are in bold.

Grasps @ action thr.	Recall↑	Precision↑	$F_{0.4}$ ↑	IoU↑
1st grasp @ 1	0.5382	**0.8134**	**0.7444**	0.4728
All grasps @ 1	**0.7394**	0.6857	0.6829	**0.5461**
1st grasp @ 2	0.1817	**0.9213**	0.5112	0.1797
All grasps @ 2	**0.4778**	0.8814	**0.7548**	**0.4471**
All grasps @ 3	**0.2269**	**0.9347**	**0.5152**	**0.2211**
All grasps @ 4	**0.1137**	**0.9791**	**0.2990**	**0.1123**

VI. Discussion and Conclusions

We have proposed the *motion correlation* (MCor) method which can segment a grasped moving cable even when the robot or the cable perturbs neighboring cables. It exploits the observation that gripper motion tends to correlate with grasped cable motion estimated in a common image frame by an optical flow predictor. We have tested MCor and the MSeg baseline on data recorded with our physical robotic setup. Both quantitative and qualitative results indicate that MCor outperforms the baseline and several passive cable segmentation methods. With one set of parameters, MCor segments well both hoses and ropes.

Once MCor partially segments a cable by moving it at a given grasp point, a grasp sampling algorithm can process the partial segmentation to propose further grasps. This allows the robot to automatically interact with the explored cable at multiple grasp points. After a sufficient number of grasps, MCor produces (almost) complete cable segmentation masks.

Limitations: Although MCor wrongly segments the robot arm significantly less often than MSeg, it can sometimes happen (see Fig. 4 b). We think it is due to MfnProb FT incorrectly estimating arm-motion consistent optical flow between the robot arm and the cables/background. Fine-tuning MfnProb FT to reject such wrong correspondences may solve the problem.

The cable segments close to the mounted endpoints are often both hard to grasp and hard to move, thus MCor does not segment them very often. However, our methods cannot completely avoid grasping them. Future work could measure the interaction force or use an endpoint detector to avoid moving or grasping endpoint segments.

Gripper-cable motion correlation cannot distinguish individual cables when multiple cables tightly interact with each other

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

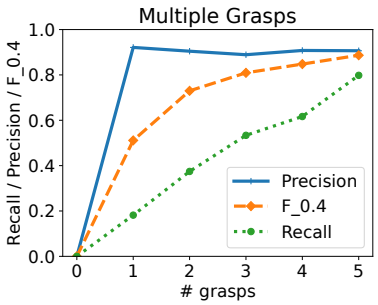

Fig. 5: Mean MCor performance with multiple grasps and the action threshold set to two.

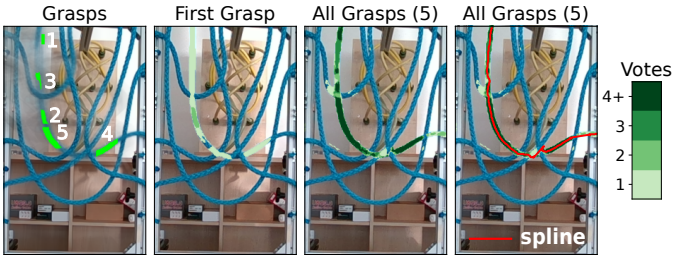

Fig. 6: Sample MCor result given one and five grasp points.

during the gripper motion (see video). If that happens, the robot may need to find an action direction which avoids the tight interactions or use two robotic arms to physically verify cable segment connectivity, e.g. by pulling two cable segments apart. We leave these possible extensions for future work.

REFERENCES

[1] A. Choi, *et al.*, "mBEST: Realtime deformable linear object detection through minimal bending energy skeleton pixel traversals," *IEEE Robot. Autom. Lett.*, vol. 8, no. 8, pp. 4863–4870, aug 2023.

[2] A. Caporali, *et al.*, "RT-DLO: Real-time deformable linear objects instance segmentation," *IEEE Trans. Ind. Informat.*, vol. 19, no. 11, pp. 11 333–11 342, nov 2023.

[3] A. Caporali, *et al.*, "FASTDLO: Fast Deformable Linear Objects Instance Segmentation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9075–9082, oct 2022.

[4] O. Holešovský, *et al.*, "MovingCables: Moving Cable Segmentation Method and Dataset," *IEEE Robotics and Automation Letters*, vol. 9, no. 8, pp. 6991–6998, Aug. 2024.

[5] R. Zanella, *et al.*, "Auto-generated Wires Dataset for Semantic Segmentation with Domain-Independence," in *2021 Int. Conf. on Computer, Control and Robotics (ICCCR)*. IEEE, jan 2021.

[6] A. Caporali, *et al.*, "Ariadne+: Deep Learning–Based Augmented Framework for the Instance Segmentation of Wires," *IEEE Trans. Ind. Informat.*, vol. 18, no. 12, pp. 8607–8617, dec 2022.

[7] M. Yan, *et al.*, "Self-Supervised Learning of State Estimation for Manipulating Deformable Linear Objects," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2372–2379, apr 2020.

[8] M. Wnuk, *et al.*, "Kinematic Multibody Model Generation of Deformable Linear Objects from Point Clouds," in *2020 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, oct 2020.

[9] J. Zhu, *et al.*, "Robotic manipulation planning for shaping deformable linear objects with environmental contacts," *IEEE Robot. Autom. Lett.*, vol. 5, no. 1, pp. 16–23, jan 2020.

[10] J. Zhu, *et al.*, "Vision-based manipulation of deformable and rigid objects using subspace projections of 2D contours," *Robotics and Autonomous Systems*, vol. 142, p. 103798, aug 2021.

[11] A. Keipour, *et al.*, "Deformable One-Dimensional Object Detection for Routing and Manipulation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4329–4336, apr 2022.

[12] V. Viswanath, *et al.*, "Disentangling Dense Multi-Cable Knots," in *2021 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, sep 2021.

[13] K. Shivakumar, *et al.*, "SGTM 2.0: Autonomously untangling long cables using interactive perception," in *2023 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, may 2023.

[14] K. Lv, *et al.*, "Learning to Estimate 3-D States of Deformable Linear Objects from Single-Frame Occluded Point Clouds," in *2023 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, may 2023.

[15] A. Caporali, *et al.*, "Deformable Linear Objects 3D Shape Estimation and Tracking From Multiple 2D Views," *IEEE Robot. Autom. Lett.*, vol. 8, no. 6, pp. 3852–3859, jun 2023.

[16] S. Zhaole, *et al.*, "A Robust Deformable Linear Object Perception Pipeline in 3D: From Segmentation to Reconstruction," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 843–850, Jan. 2024.

[17] A. Caporali, *et al.*, "Dlo perceiver: Grounding large language model for deformable linear objects perception," *IEEE Robotics and Automation Letters*, vol. 9, no. 12, pp. 11 385–11 392, Dec. 2024.

[18] P. Sundaresan, *et al.*, "Untangling Dense Non-Planar Knots by Learning Manipulation Features and Recovery Policies," in *Robotics: Science and Systems XVII*. Robotics: Science and Systems Foundation, jul 2021.

[19] V. Viswanath, *et al.*, "HANDLOOM: Learned Tracing of One-Dimensional Objects for Inspection and Manipulation," in *7th Conf. on Robot Learning (CoRL)*, ser. Proc. of Machine Learning Research, J. Tan, *et al.*, Eds., vol. 229. PMLR, 06–09 Nov 2023, pp. 341–357.

[20] X. Zhang, *et al.*, "Learning Efficient Policies for Picking Entangled Wire Harnesses: An Approach to Industrial Bin Picking," *IEEE Robot. Autom. Lett.*, vol. 8, no. 1, pp. 73–80, jan 2023.

[21] C. Chi and D. Berenson, "Occlusion-robust Deformable Object Tracking without Physics Simulation," in *2019 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, nov 2019.

[22] Y. Wang, *et al.*, "Tracking Partially-Occluded Deformable Objects while Enforcing Geometric Constraints," in *2021 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, may 2021.

[23] J. Xiang, *et al.*, "TrackDLO: Tracking Deformable Linear Objects Under Occlusion With Motion Coherence," *IEEE Robot. Autom. Lett.*, vol. 8, no. 10, pp. 6179–6186, oct 2023.

[24] J. Bohg, *et al.*, "Interactive Perception: Leveraging Action in Perception and Perception in Action," *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1273–1291, dec 2017.

[25] C. J. Tsikos and R. K. Bajcsy, "Segmentation via manipulation," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 306–319, jun 1991.

[26] J. Kenney, *et al.*, "Interactive segmentation for manipulation in unstructured environments," in *2009 IEEE Int. Conf. on Robotics and Automation*. IEEE, may 2009.

[27] C. D. Singh, *et al.*, "NudgeSeg: Zero-Shot Object Segmentation by Repeated Physical Interaction," in *2021 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, sep 2021.

[28] W. Boerdijk, *et al.*, "Self-Supervised Object-in-Gripper Segmentation from Robotic Motions," in *4th Conf. on Robot Learning (CoRL 2020)*, Feb. 2020.

[29] T. Patten, *et al.*, "Action Selection for Interactive Object Segmentation in Clutter," in *2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, oct 2018.

[30] A. Eitel, *et al.*, "Self-supervised Transfer Learning for Instance Segmentation through Physical Interaction," in *2019 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, nov 2019.

[31] A. Price, *et al.*, "Fusing RGBD Tracking and Segmentation Tree Sampling for Multi-Hypothesis Volumetric Segmentation," in *2021 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, may 2021.

[32] X. Fang, *et al.*, "Embodied Uncertainty-Aware Object Segmentation," in *2024 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, oct 2024.

[33] Y. Lu, *et al.*, "Self-Supervised Unseen Object Instance Segmentation via Long-Term Robot Interaction," in *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation, jul 2023.

[34] H. H. Qian, *et al.*, "RISeg: Robot Interactive Object Segmentation via Body Frame-Invariant Features," in *2024 IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, May 2024, pp. 13 954–13 960.

[35] M. Stoiber, *et al.*, "A multi-body tracking framework - from rigid objects to kinematic structures," 2022.

[36] N. Ravi, *et al.*, "SAM 2: Segment Anything in Images and Videos," *arXiv*, Aug. 2024.