2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

CLIPUNetr: Assisting Human-robot Interface for Uncalibrated Visual Servoing Control with CLIP-driven Referring Expression Segmentation

Chen Jiang†, Yuchen Yang† and Martin Jagersand†

***Abstract*— The classical human-robot interface in uncalibrated image-based visual servoing (UIBVS) relies on either human annotations or semantic segmentation with categorical labels. Both methods fail to match natural human communication and convey rich semantics in manipulation tasks as effectively as natural language expressions. In this paper, we tackle this problem by using referring expression segmentation, which is a prompt-based approach, to provide more in-depth information for robot perception. To generate high-quality segmentation predictions from referring expressions, we propose CLIPUNetr - a new CLIP-driven referring expression segmentation network. CLIPUNetr leverages CLIP's strong vision-language representations to segment regions from referring expressions, while utilizing its "U-shaped" encoder-decoder architecture to generate predictions with sharper boundaries and finer structures. Furthermore, we propose a new pipeline to integrate CLIPUNetr into UIBVS and apply it to control robots in real-world environments. In experiments, our method improves boundary and structure measurements by an average of 120% and can successfully assist real-world UIBVS control in an unstructured manipulation environment.**

I. Introduction

Uncalibrated Image-Based Visual Servoing (UIBVS) [1] is a well-established method to enact position-based robot control. The pipeline of UIBVS can be broadly divided into two components: 1) perception, and 2) control. The perception phrase uses visual algorithms to analyze the environment from camera inputs, and to extract low-level image geometric features. The control phase takes in the geometric features and performs visuo-motor control. Originally, perception in UIBVS requires intense human labor for interfacing, in which the users have to review the scene in person and manually select the categories or shapes of the target [1]. More recent studies, integrated with AI [2]–[5], capture the semantics of the robot environment by training fix-class segmentation models. Though this process successfully reduces human involvement by interacting with specific categorical names in the human-robot interface, it still diverges from natural human communication. Predefined categories are inherently limited in encapsulating the rich, nuanced semantics of objects in manipulation tasks. As a result, they often fail to convey the depth of information that natural language expressions can offer.

Recent developments of large-scale pretrained vision-language models (VLMs) like CLIP [6], enabled strong vision-language representation trained from large-scale internet data. These advancements have given rise to methods such as CLIPSeg [7], capable of segmenting regions by descriptive texts. These methods have inspired us, highlighting a promising approach where the melding of rich language expression representation and image segmentation can potentially surpass the capabilities of traditional image-only semantic models, particularly in delineating more accurate and nuanced boundaries and structures. Motivated by this, we intend to further explore this hypothesis, aiming to develop a more intuitive and human-like communication approach by incorporating referring expression segmentation in the interfacing of UIBVS.

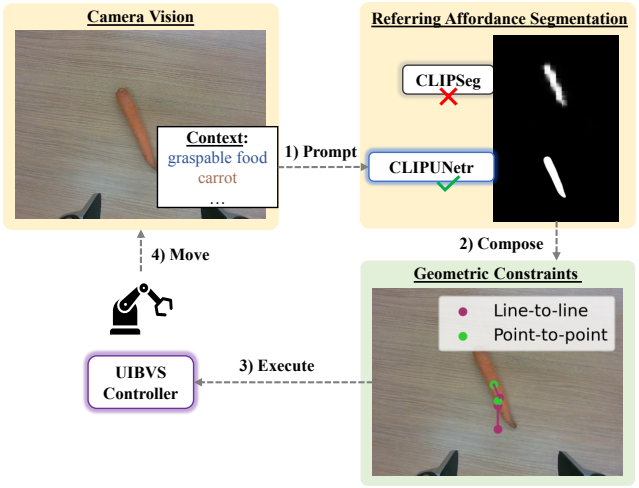

Fig. 1: Overview of our method. Given visual inputs, CLIPUNetr receives a referring expression, which is prompt-based, from a user to segment regions. Then, geometric constraints are composed from the prediction to execute UIBVS and move the robot.

In detail, we introduce CLIPUNetr[1], a network adept at segmenting images into regions by leveraging user-specified referring expressions — sentence prompts that vividly describe a target object — alongside captured images. CLIPUNetr distinguishes itself by not only delivering precise segmentation results, marked by fine detailing in boundaries and structures, but also by mitigating the necessity for the laborious clicking traditionally associated with UIBVS interfaces and overcoming the limitations ingrained in category-specific approaches. Furthermore, we pioneer a new pipeline that seamlessly integrates CLIPUNetr as a perception module within the UIBVS framework. This incorporation amplifies

†Authors are with Department of Computing Science, University of Alberta, Edmonton AB., Canada, T6G 2E8. {cjiang2, yy17, mj7}@ualberta.ca

[1]Our code is available at: https://github.com/cjiang2/clipunetr.

the perception module's ability to adeptly handle affordances and other rich semantics during manipulation tasks, fostering a more intuitive and effective communication between users and the robot control system. A summary of our method can be seen in Figure 1. We summarize our contributions as follows:

- We introduce a new referring expression based segmentation model, CLIPUNetr. The model is capable of capturing the multi-scale information through its "U-shaped" encoder-decoder, and utilizing the vision-language representations from CLIP.
- We propose a new, modulated pipeline to perform UIBVS with prompt-based robot perception. We integrate CLIPUNetr as a perception module in this pipeline, enabling users to specify prompts for composing geometric motor skills.

II. Related Work

A. Image Segmentation and Foundation Models

There have been various studies on image segmentation applied in various sub-fields, like salient object segmentation, affordance segmentation, etc. Models like BASNet [8], U^2Net [9] and DISNet [10] learn to encode multi-scale information. While the models achieve high spatial accuracy in boundary and structure, their predictions are fixed classes, and therefore restricted by categorical annotations. With the availability of pretrained models like vision transformers [11] and CLIP [6], modern segmentation methods tend to use those large pretrained models as foundation models and finetune customized decoders on downstream tasks. Models like CLIPSeg [7], CRIS [12] and LSeg [13] leveraged image-text representations from CLIP for language-driven segmentation tasks, while models like SegViT [14] and UNETR [15] inferred attention masks from ViT with transformer decoders to generate segmentation results. However, evaluation only attends to regional accuracy (e.g. IoU), while spatial accuracy of boundary (e.g. S-measure [16]) is ignored.

B. Robot Perception and Affordance

One traditional way to construct robot perception is to train vision models and perform image segmentation to predict affordances for robot control. In Do et al [3] and Chu et al [17], correct robot commands were inferred from affordances predicted by semantic models. K-VIL [18] processed keypoints from demonstrations to form geometric constraints usable by imitation learning agents. Other methods [5], [19], [20] combined affordance and keypoint predictions to guide robot control. With the introduction of language as control commands, methods like Nguyen et al [21] and Yang et al [22], [23] combined affordance prediction and vision-language models to construct modulated controllers that interface with language commands. However, training of the affordance segmentation models is expensive, and usually requires a large amount of annotated affordance data. Recently, foundation models like CLIP are utilized to train language-conditioned visuo-motor control policies. In this case, models are usually constructed as keypoint prediction modules like transporter [24], and affordances are learned unsupervised from demonstrations. Typical models include CLIPort [25] and Perceiver-actor [26], where CLIP is used to encode actionable language commands, and affordances are represented as keypoints to guide the policy learning. However, the predicted affordances are less defined and structured versus the traditional approach.

III. Methodology

A. Referring Expressions in Manipulation Workspace

Given a referring expression as a prompt, and an image of a manipulation workspace with one or more objects, the goal is to segment the pixels that best match the expression. To refer to objects in the manipulation workspace in a contextual manner, the prompt describes important semantics like object naming, shape, functionality or affordances. Sample templates of prompts are visualized in Figure 2, and categorized as follows:

- **Affordance-enriched Prompts** The prompt describes the functionality of an object, or parts of an object, in diverse, descriptive expressions.
- **Object-oriented Prompts** The prompt refers to an object explicitly by its common naming, or implicitly by describing shapes, colors, or intentional usages.

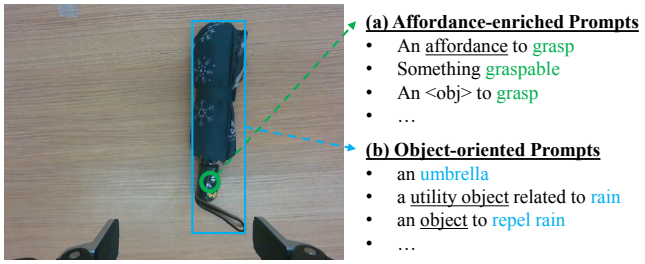

Fig. 2: Two types of prompts to refer to an umbrella object.

The two types of prompts can be combined to acquire prompts containing both affordance and object-oriented descriptions.

B. Architecture

The proposed CLIPUNetr, visualized in Figure 3, is a U-Net-like Encoder-Decoder network, which encodes the input image-prompt pairs, and decodes the segmentation probability maps.

Visual Encoder First, CLIPUNetr infers CLIP's pretrained visual encoder. The encoder is a vision transformer (ViT) which takes an image input I to capture semantic information. The image is first divided into N patch embeddings E_I, where $N = HW/r^2$, r is the patch resolution, H and W are the image height and width. A learnable class embedding I_{cls} is concatenated with the patch embeddings, acquiring the token as $\{I_{cls}; E_I\}$, and the learnable positional embedding E_{pos} is added to the token. Then, the token is processed by L Transformer layers, composed by alternating multi-headed self-attention (MSA) and MLP blocks. The token from the

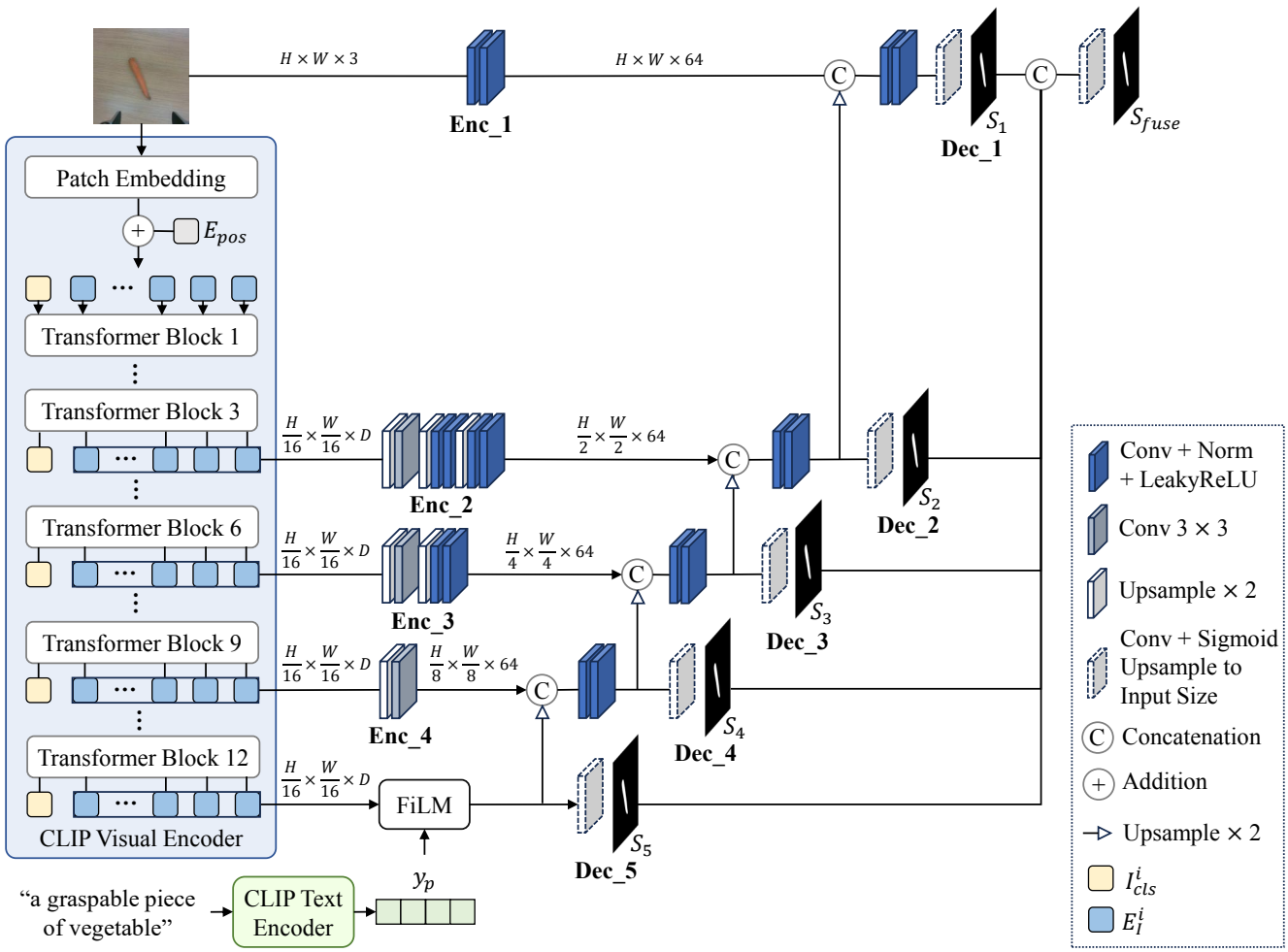

Fig. 3: Architecture of CLIPUNetr. Visual encoder encodes the image and generates token features and encoder features. The last token feature is conditioned with the text embeddings of the prompt through FiLM. The decoders take the conditioned token features and the encoder features to generate 5 side outputs, which are fused into a single segmentation probability map.

final Transformer layer is then processed by an embedding head, outputting the image embedding y_I:

$$\begin{aligned} E_I^j &= PatchEmbed(I^j), j = 1, 2, ..., N \\ [I_{cls}^0; E_0] &= [I_{cls}; E_I] + E_{pos} \\ [I_{cls}^{i+1}; E_I^{i+1}] &= Layer^i([I_{cls}^i; E_I^i]), 0 \leq i < L-1 \\ y_I &= Head(I_{cls}^L) \end{aligned} \tag{1}$$

where $Layer^i$ denotes the i^{th} transformer block. To support images with higher resolution, we interpolate I_{cls} as discussed in Dosovitskiy et al [11].

Then, similar to UNETR [15], additional encoders are attached, which take the same image input I, and tokens from the CLIP visual encoder. For image I, one block of 3×3 convolution, leaky ReLU and normalization layers are applied. For the token $\{I_{cls}^i; E_I^i\}$, where $i \in \{3, 6, 9, 12\}$, the class embedding I_{cls}^i is removed, and E_I^i is reshaped from $N \times D$ to $r \times r \times D$. The reshaped embedding feature is then upsampled by a scale factor of 2, followed by consecutive 3×3 convolution, leaky ReLU and normalization layers for a number of times.

Feature-wise Linear Modulation To inform the decoder with information from the prompt, the last encoder feature is conditioned with text embeddings through Feature-wise Linear Modulation (FiLM) [27]. Given a prompt p, CLIP text encoder encodes the prompt with transformer blocks, outputting a category [CLS] token T_{cls}. The token is projected by an embedding head to generate the text embedding y_p. For the token E_I^L from L^{th} layer, FiLM learns an affine transformation to perform conditional scaling:

$$FiLM(E_I^L) = \gamma(y_p)E_I^L + \beta(y_p) \tag{2}$$

where γ and β are learnable parameters.

Segmentation Decoder To generate the resulting probability maps with high resolution, we enable feature scaling with side outputs in the construction of CLIPUNetr's U-shaped segmentation decoder, inspired by BASNet [8]. First,

the language-conditioned token $FiLM(E_I^L)$ is upsampled with a scale factor of 2, followed by consecutive 3 $\times$ 3 convolution, leaky ReLU and normalization layers. Then, the output is upsampled and merged with the feature of the previous transformer output via skip connections. The concatenated feature is again processed by another consecutive 3 $\times$ 3 convolution, leaky ReLU and normalization layers. The process is repeated till the output features are upsampled to the original input resolution, acquiring 5 decoding features. Then, we produce side output probability maps, and refine the side outputs to generate the final probability map. Each decoder feature will be fed into a 3 $\times$ 3 convolution, followed with upsampling and a sigmoid function. The process generates 5 side output probability maps S_5, S_4, S_3, S_2, S_1. The side outputs are concatenated and refined by a 1×1 convolution and a sigmoid function, generating the final probability map S_{fuse}.

Hybrid Loss Similar to BASNet [8], we define loss as the summation of all outputs:

$$\mathcal{L} = \alpha_{fuse}\ell_{fuse} + \sum_{k=1}^{5}(\alpha_{side}^{k}\ell_{side}^{k}) \tag{3}$$

where ℓ_{fuse} is the loss for fused probability map, ℓ_{side}^{k} is the loss for side probability map, α_{fuse} and α_{side}^{k} are the weights of each loss term . Each individual loss term ℓ is calculated by a hybrid loss, which is composed by the summation of binary cross entropy, SSIM and IoU loss:

$$\ell = \ell_{bce} + \ell_{ssim} + \ell_{iou} \tag{4}$$

Utilizing a hybrid loss forces the decoder to focus on the foreground and preserve the structure of the predicted segmentation results, especially near the boundary.

C. Geometric Constraint Composition

To align the robot end effector position to a target position in the manipulation workspace, we construct geometric constraints [1] from image geometry, which specify the alignment context. In detail, given a list of ordered keypoints $\mathbf{f} = \{f_1, f_2, ..., f_n\}$ in homogeneous coordinates, a geometric constraint T is a task function that maps the keypoints into $\{0, 1\}$, $T(f) = 0$ when the robot configuration is aligned with the task description, $T(f) = 1$ when the alignment is violated (e.g. unparalleled lines). In the scope of this paper, four basic types of task descriptions are considered:

$$\begin{aligned} T_{pp}(\mathbf{f}) &= f_2 - f_1 \\ T_{pl}(\mathbf{f}) &= f_1 \cdot l_{23} \\ T_{ll}(\mathbf{f}) &= f_1 \cdot l_{34} + f_2 \cdot l_{34} \\ T_{par}(\mathbf{f}) &= l_{12} \times l_{34} \end{aligned} \tag{5}$$

where T_{pp}, T_{pl}, T_{ll}, and T_{par} are denoted as point-to-point (p2p), point-to-line (p2l), line-to-line (l2l), and parallel-line (par) task, respectively. A line l_{ij} is denoted by the cross product of two points f_i and f_j.

Given the output probability map M from CLIPUNetr and the prompt p, the process to compose geometric constraint in an eye-in-hand camera configuration is expanded as follows: First, the probability map is used as an image descriptor function $M(F) = (M(f_1), M(f_2), ..., M(f_k), ..., M(f_K))$, where $M(f_k)$ denotes the probability score of p at a keypoint location f_k, $F = \{f_1, f_2, ..., f_k, ..., f_K\}$ defines the full image grid, where $K = HW$. Next, by filtering the probability score with a threshold of 0.5, we obtain a set of candidate keypoints $F_{candidates}$, where $f_k \in F_{candidates}$ and $M(f_k) > 0.5$. Then, PCA is used to analyze the variance of $F_{candidates}$, obtaining the principal point f_{point} and principal lines f_{line}. Last, simple heuristics are used to compose the constraints, visualized in Figure 4. For a p2p task, the target position $f_2 = f_{point}$ aligns to the end effector position f_1, heuristically set as $f_1 = (W/2, 4H/5, 1)$. Similarly for a p2l task, the end effector position f_1 is set as $(W/2, 4H/5, 1)$, while the target line l_{23} is set as f_{line}. For l2l and par task, the end effector orientation l_{12} is defined as the vertical line passing through mid image center, while the target line l_{34} is set as f_{line}.

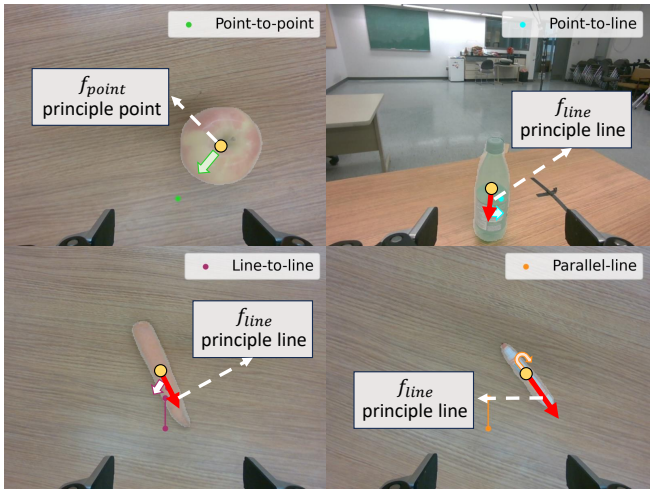

Fig. 4: Heuristics to compose four types of geometric constraints from the predictions of CLIPUNetr.

D. UIBVS with CLIPUNetr in Perception

UIBVS [28] defines the visual-motor control law, where a robot can be controlled to reach a desired joint configuration from image geometric inputs. The equation is denoted as:

$$\dot{e} = J_u(q)\dot{q} \tag{6}$$

where $\dot{q}$ is the control input of a robot with N joints, J_u is the visuo-motor Jacobian. The values of the geometric constraints, calculated by combining heuristics and referring expression segmentation, are used as the error signal $\dot{e}$. Then, Broyden update is performed in replacement of camera calibration and analytical Jacobian calculation:

$$\hat{J}_u^{(k+1)} = \hat{J}_u^{(k)} + \lambda\frac{(e - \hat{J}_u^{(k)}\Delta q)\Delta q^T}{\Delta q^T \Delta q + \epsilon} \tag{7}$$

where λ is the weight of the rank one Broyden update.

Following the above definitions, Algorithm 1 shows our pipeline to interface UIBVS with CLIPUNetr. Given an image observation I_t at time t, the human operator first decides the geometric constraints to compose, and then specifies a prompt p to initiate the robot perception. For each I_t, CLIPUNetr will be inferred, generating the probability map M_t. m geometric constraints are composed following the discussed strategy, and are inputted into the UIBVS controller, generating the joint commands to move the robot.

Algorithm 1: Robot perception to compose geometric constraints and perform UIBVS control.

Inputs: A randomly sampled image frame I_t. Geometric constraints, text prompt p, specified by a human operator.
Result: Joint command ΔJ.
while *True* **do**
  sample I_t;
  M_t = CLIPUNetr(I_t, p);
  f_{point}, f_{line} = PCA(M_t);
  $\{T_1, ..., T_m\}$ = ComposeConstraint(f_{point}, f_{line});
  ΔJ = UIBVS($\{T_1, ..., T_m\}$);
end

IV. Experiments

A. Experimental Setup

Datasets We use the publicly available PhraseCut and UMD+GT datasets for experiments. PhraseCut dataset [29] contains 340,000 phrases with associating regional referring expression prompts. UMD+GT dataset [5] contains 30,000 RGBD images of 104 objects, originally annotated with 6 fix-class affordance labels. To generate referring expressions for UMD+GT dataset, we manually re-annotate the images based on 39 diverse prompt templates. In training, we randomly sample one of the prompts, and in testing, we replace the affordance label with one object-oriented prompt, and one affordance-enriched prompt. Prediction is done twice and we take the average metric scores of the two predictions. To evaluate object boundary and structure on both datasets, we use four metrics studied in salient object segmentation: Mean Absolute Error (MAE), Structure measure, weighted F-measure, and max F-measure.

Robot Control The robot control is evaluated offline and online. In offline setup, 19 robot manipulation tasks of moving and grasping, controlled by the classical manual clicking interface [1] and performed by a Kinova Gen3, are recorded using an eye-in-hand Intel Realsense D405 camera. Masks are annotated and evaluated every 10^{th} frame.

In online setup, CLIPUNetr is integrated as a perception module to replace visual tracking with manual clicking. We compare the modified interface against the classical interface [1] by completing 12 robot manipulation tasks of moving and grasping with 5 food objects (apple, red pepper, lemon, carrot, banana), 2 utility objects (tennis ball, umbrella) and 4 marker pens. 3 attempts are allowed for the user to complete a task. The task is successful if the robot approaches and grasps the object without falling. We report the success rate for the three categories of tasks.

Implementation Details The model is implemented using PyTorch, using the same hyperparameters from CLIPSeg [7]. A batch size of 32 is used to train the models on a single Nvidia V100 GPU. For UIBVS control, joint 1, 2, 6 and 7 are used for table-top manipulation, and the implementation is done in ROS. A λ of 0.05 is set for Broyden's update. Joint angle commands are published in 1 Hz to move the robot end effector in small amounts, until convergence.

B. Results for Referring Expression Segmentation

TABLE I: Quantitative results for referring expression segmentation.

Dataset	Model	MAE↓	$S_m \uparrow$	$wF_\beta \uparrow$	$maxF_\beta \uparrow$
PhraseCut	CLIPUNetr	**0.0840**	**0.6975**	**0.5375**	**0.5887**
	CLIPSeg [11]	0.1343	0.6629	0.3836	0.5658
	CLIPSeg*	0.1098	0.6865	0.4263	0.5840
UMD+GT	CLIPUNetr	**0.0025**	**0.8968**	**0.7770**	0.8021
	CLIPSeg [11]	0.1714	0.5447	0.1165	0.3335
	CLIPSeg*	0.0064	0.7858	0.4705	0.6027
	AffKp [5]	0.0044	0.8756	0.6753	**0.8030**

Quantitative Evaluation Table I shows the quantitative results of referring expression segmentation on PhraseCut and UMD+GT datasets. For fair comparison, an reimplementation of CLIPSeg is trained following our training regime (CLIPSeg*). In summary, CLIPUNetr is able to outperform CLIPSeg on both PhraseCut dataset and UMD+GT dataset, reflecting CLIPUNetr's capability of generating good quality predictions. For UMD+GT dataset, CLIPUNetr surpasses the performance of AffKp, a supervised network trained only with fix-class labels. This proves that leveraging image-text representations from CLIP is beneficial for learning.

Qualitative Evaluation To further illustrate the superior performance of CLIPUNetr, Figure 5 visualizes the prediction results from CLIPUNetr vs. CLIPSeg. As shown, predictions from CLIPSeg contain checkerboard artifacts, caused by abruptly upsampling low-resolution decoding features in the decoder. This validates the effectiveness of CLIPUNetr to learn multi-scale information, thus acquiring predictions with more accurate object boundaries and finer structures.

C. Ablation Study

We validate the effectiveness of each key component used in CLIPUNetr in two parts: architecture ablation and loss ablation. Table II presents the results.

Architecture Ablation We take CLIPSeg as the base network, and first extend its decoder to use the same features from layer 3, 6, 9 and 12, denoted as CLIPSeg-NoSides. Then, we extend the decoder with side outputs, denoted as CLIPSeg-Sides. As shown, the model with side outputs achieves superior results.

Loss Ablation We take CLIP-Deconv as the base network, where simple MLP and deconvolution layers are inferred to decode segmentation outputs. CLIP-Deconv is first trained

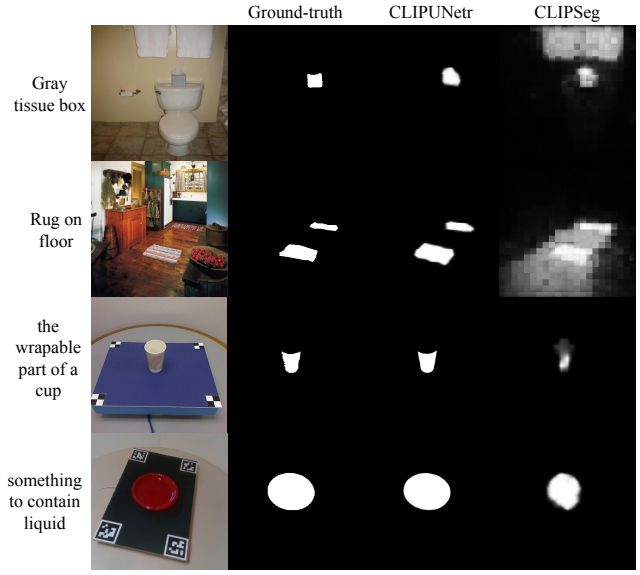

Fig. 5: Qualitative results on PhraseCut dataset and UMD+GT dataset.

TABLE II: Ablation study on architecture and losses.

Dataset	Model	MAE↓	$S_m \uparrow$	$wF_\beta \uparrow$	$maxF_\beta \uparrow$
PhraseCut	CLIPUNetr	**0.0840**	0.6975	**0.5375**	**0.5887**
	CLIPSeg-Sides	0.0844	**0.6989**	0.5326	0.5883
	CLIPSeg-NoSides	0.1240	0.6083	0.3779	0.4252
	CLIP-Deconv	0.0903	0.6668	0.4683	0.5356
	CLIP-Deconv-BCE	0.1319	0.6468	0.3511	0.5489
UMD+GT	CLIPUNetr	**0.0025**	**0.8968**	**0.7770**	**0.8021**
	CLIPSeg-Sides	0.0039	0.8347	0.6707	0.6895
	CLIPSeg-NoSides	0.0056	0.7934	0.5450	0.5908
	CLIP-Deconv	0.0035	0.8285	0.6694	0.6932
	CLIP-Deconv-BCE	0.0082	0.7772	0.4209	0.5873

with only binary cross entropy, denoted as CLIP-Deconv-BCE. Then, the training is extended with the hybrid loss, denoted as CLIP-Deconv. In summary, CLIP-Deconv achieves superior results, validating the fact that using hybrid loss is beneficial in learning structure information.

D. Results for Robot Control

Offline Evaluation Table III presents the quantitative results of CLIPUNetr with robot data. Our CLIPUNetr achieves superior performance versus CLIPSeg, being able to handle complex motions in unseen robot demonstration videos. Additionally, qualitative comparisons are presented in Figure 6. Versus CLIPSeg, where predictions are plagued by checkerboard artifacts, CLIPUNetr is able to generate predictions with good quality object boundary and fine structure. This ensures the viability to use CLIPUNetr for robot perception. Additionally, we attach the augmented prompts used to generate the segmentation results. Thanks to the prompt templates used in training, CLIPUNetr is able to better handle diverse prompts containing both affordance and object information. This allows the users to specify prompts in diverse ways, increasing the flexibility of the interface.

Online Evaluation The success rate of robot control over three categories of objects are presented in Table IV. In summary, perception with CLIPUNetr shows comparable

TABLE III: Quantitative results with robot data.

Model	MAE↓	$S_m \uparrow$	$wF_\beta \uparrow$	$maxF_\beta \uparrow$
CLIPUNetr	**0.0101**	**0.9049**	**0.8404**	**0.9104**
CLIPSeg [11]	0.0374	0.8075	0.4720	0.7736

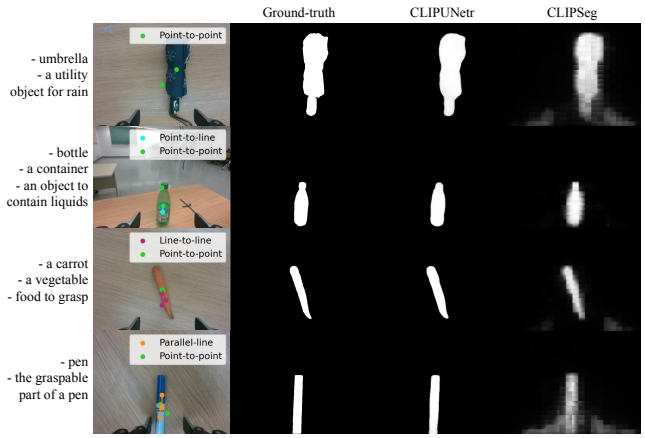

Fig. 6: Qualitative results with robot data.

performance versus the classical visual interface, being able to successfully segment the unseen daily living objects and perform moving and grasping actions over them. For classical interface, 1 out of 12 tasks fails as a result of poor visual tracking performance. With CLIPUNetr in robot perception, manual clicking can be successfully avoided. And the robot can successfully utilize results of referring expression segmentation to compose geometric constraints and complete the designated manipulation tasks.

TABLE IV: Average success rate of the robot manipulation tasks.

Name	Category	Success Rate
w/ CLIPUNetr	Food	100%
	Marker Pen	100%
	Utility	100%
Classical [1]	Food	80%
	Marker Pen	100%
	Utility	100%

V. Conclusions

In this paper, we enhance the human-robot interface of UIBVS with referring expressions. First, we construct CLIPUNetr, where the network adapts CLIP, feature-wise linear modulation and feature scaling with side outputs to perform referring expression segmentation. Second, we build a pipeline to integrate CLIPUNetr into the robot perception and conduct UIBVS control. In comparison, CLIPUNetr improves object boundary and structure measurements by an average of 120%, producing results with sharper boundaries and finer structures. Integrated into the robot perception, CLIPUNetr can successfully segment objects in an unstructured workspace and assist with UIBVS control. Promising lines of future work include improvements in inference speed, and adaptations of the model in assistive robotics.

REFERENCES

[1] M. Gridseth, O. Ramirez, C. P. Quintero, and M. Jagersand, "Vita: Visual task specification interface for manipulation with uncalibrated visual servoing," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3434–3440.

[2] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos, "Affordance detection of tool parts from geometric features," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1374–1381.

[3] T.-T. Do, A. Nguyen, and I. Reid, "Affordancenet: An end-to-end deep learning approach for object affordance detection," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 5882–5889.

[4] B. Griffin, V. Florence, and J. Corso, "Video object segmentation-based visual servo control and object depth estimation on a mobile robot," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 1647–1657.

[5] R. Xu, F.-J. Chu, C. Tang, W. Liu, and P. A. Vela, "An affordance keypoint detection network for robot manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2870–2877, 2021.

[6] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[7] T. Lüddecke and A. Ecker, "Image segmentation using text and image prompts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7086–7096.

[8] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "Basnet: Boundary-aware salient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 7479–7489.

[9] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, "U2-net: Going deeper with nested u-structure for salient object detection," *Pattern recognition*, vol. 106, p. 107404, 2020.

[10] X. Qin, H. Dai, X. Hu, D.-P. Fan, L. Shao, and L. Van Gool, "Highly accurate dichotomous image segmentation," in *European Conference on Computer Vision*. Springer, 2022, pp. 38–56.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[12] Z. Wang, Y. Lu, Q. Li, X. Tao, Y. Guo, M. Gong, and T. Liu, "Cris: Clip-driven referring image segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 686–11 695.

[13] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," 2022. [Online]. Available: https://openreview.net/forum?id=RriDjddCLN

[14] B. Zhang, Z. Tian, Q. Tang, X. Chu, X. Wei, C. Shen *et al.*, "Segvit: Semantic segmentation with plain vision transformers," *Advances in Neural Information Processing Systems*, vol. 35, pp. 4971–4982, 2022.

[15] A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, A. Myronenko, B. Landman, H. R. Roth, and D. Xu, "Unetr: Transformers for 3d medical image segmentation," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2022, pp. 574–584.

[16] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4548–4557.

[17] F.-J. Chu, R. Xu, C. Tang, and P. A. Vela, "Recognizing object affordances to support scene reasoning for manipulation tasks," *arXiv preprint arXiv:1909.05770*, 2019.

[18] J. Gao, Z. Tao, N. Jaquier, and T. Asfour, "K-vil: Keypoints-based visual imitation learning," *IEEE Transactions on Robotics*, 2023.

[19] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "kpam: Keypoint affordances for category-level robotic manipulation," in *The International Symposium of Robotics Research*. Springer, 2019, pp. 132–157.

[20] W. Gao and R. Tedrake, "kpam 2.0: Feedback control for category-level robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2962–2969, 2021.

[21] A. Nguyen, D. Kanoulas, L. Muratore, D. G. Caldwell, and N. G. Tsagarakis, "Translating videos to commands for robotic manipulation with deep recurrent neural networks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3782–3788.

[22] S. Yang, W. Zhang, W. Lu, H. Wang, and Y. Li, "Learning actions from human demonstration video for robotic manipulation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1805–1811.

[23] S. Yang, W. Zhang, R. Song, J. Cheng, H. Wang, and Y. Li, "Watch and act: Learning robotic manipulation from visual demonstration," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.

[24] T. D. Kulkarni, A. Gupta, C. Ionescu, S. Borgeaud, M. Reynolds, A. Zisserman, and V. Mnih, "Unsupervised learning of object keypoints for perception and control," *Advances in neural information processing systems*, vol. 32, 2019.

[25] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 894–906.

[26] ——, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 785–799.

[27] V. Dumoulin, E. Perez, N. Schucher, F. Strub, H. d. Vries, A. Courville, and Y. Bengio, "Feature-wise transformations," *Distill*, 2018, https://distill.pub/2018/feature-wise-transformations.

[28] M. Jagersand, O. Fuentes, and R. Nelson, "Experimental evaluation of uncalibrated visual servoing for precision manipulation," in *Proceedings of International Conference on Robotics and Automation*, vol. 4. IEEE, 1997, pp. 2874–2880.

[29] C. Wu, Z. Lin, S. Cohen, T. Bui, and S. Maji, "Phrasecut: Language-based image segmentation in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 216–10 225.