MROSS: Multi-Round Region-based Optimization for Scene Sketching

Yiqi Liang, Liu Ying*, Dandan Long, Ruihui Li
College of Computer Science and Electronic Engineering
Hunan University
Changsha, China
yiqiliang@hnu.edu.cn, liu_ying@hnu.edu.cn, ddlong@hnu.edu.cn, liruihui@hnu.edu.cn

***Abstract*—Scene sketching is to convert a scene into a simplified, abstract representation that captures the essential elements and composition of the original scene. It requires a semantic understanding of the scene and consideration of different regions within the scene. Since scenes often contain diverse visual information across various regions, such as foreground objects, background elements, and spatial divisions, dealing with these different regions poses unique difficulties. In this paper, we define a sketch as some sets of Bézier curves because of their smooth and versatile characteristics. We optimize different regions of input scene in multiple rounds. In each optimization round, the strokes sampled from the next region can seamlessly be integrated into the sketch generated in the previous optimization round. We propose an additional stroke initialization method to ensure the integrity of the scene and the convergence of optimization. A novel CLIP-based Semantic Loss and a VGG-based Feature Loss are utilized to guide our multi-round optimization. Extensive experimental results on the quality and quantity of the generated sketches confirm the effectiveness of our method.**

***Index Terms*—sketch, vector image, optimization, image processing**

I. Introduction

Scene sketching refers to the process of creating rough sketches or drawings to visually represent a scene or environment. It is commonly used in various fields, including art, design, architecture, film, and animation.

Scene sketching offers advantage of conveying information quickly and efficiently. It provides a concise visual summary that allows viewers to grasp the overall layout, spatial relationships, and key features at a glance. Additionally, scene sketching facilitates the creative process by enabling artists and designers to visualize their ideas and concepts [1]–[3].

However, generating a scene sketch is highly challenging, as it requires the ability to understand and depict the visual characteristics of the scene with complex subjects and interactions. [3]–[6] often rely on explicit sketch datasets for training. The sketches of them are often simplified and abstract expressions of the original images, with a fixed style or preset. It is difficult to balance visual effect of sketches, producing visual appeal and aesthetics.

Besides, different regions within a scene may have varying levels of importance or prominence as seen in Fig. 1. For example, foreground objects or focal points might require more attention to detail and precision in sketching, while background elements may be more loosely represented. Some works can achieve this with flexibility. However, these works focus specifically on the task of object sketching [7], [8] or portrait sketching [9], [10], and often simply use the number of strokes to define the effect of their sketches.

Fig. 1. Drawings of different scenes by different artists. Notice the significant differences in level of abstraction between different regions of the drawings.

To address the above two issues, we introduce a scene sketching method based on regions with multi-round optimization. We utilize the black parametric Bézier curves as our fundamental shape primitive for strokes of a sketch and optimized them by a pre-trained CLIP-ViT model [11], [12] and VGG16 model [13].

Unlike [14]–[16], optimization is performed for different regions in our method, which helps to highlight regions of interest, achieving coarse-to-concrete sketches. An intuitive and succinct learning process can be seen in Fig. 2. In each round of optimization, we are in pursuit of full content exploration rather than only the salient guidance. To achieve this, we present an edge-based stroke initialization and utilize a farthest point sampling (FPS) algorithm to uniformly sample strokes in the input image. Fig. 5 shows that our sampling method has a more efficient effect and is able to generate sketches in a reasonable manner. To transform a detailed scene into a simplified sketch, one must condense intricate visual elements into fundamental lines, shapes, and tones, all while retaining the scene's identifiable characteristics. We utilize intermediate layer of CLIP-ViT [12] to guide the optimization, where encourages the creation of looser sketches that emphasize

* Corresponding author.

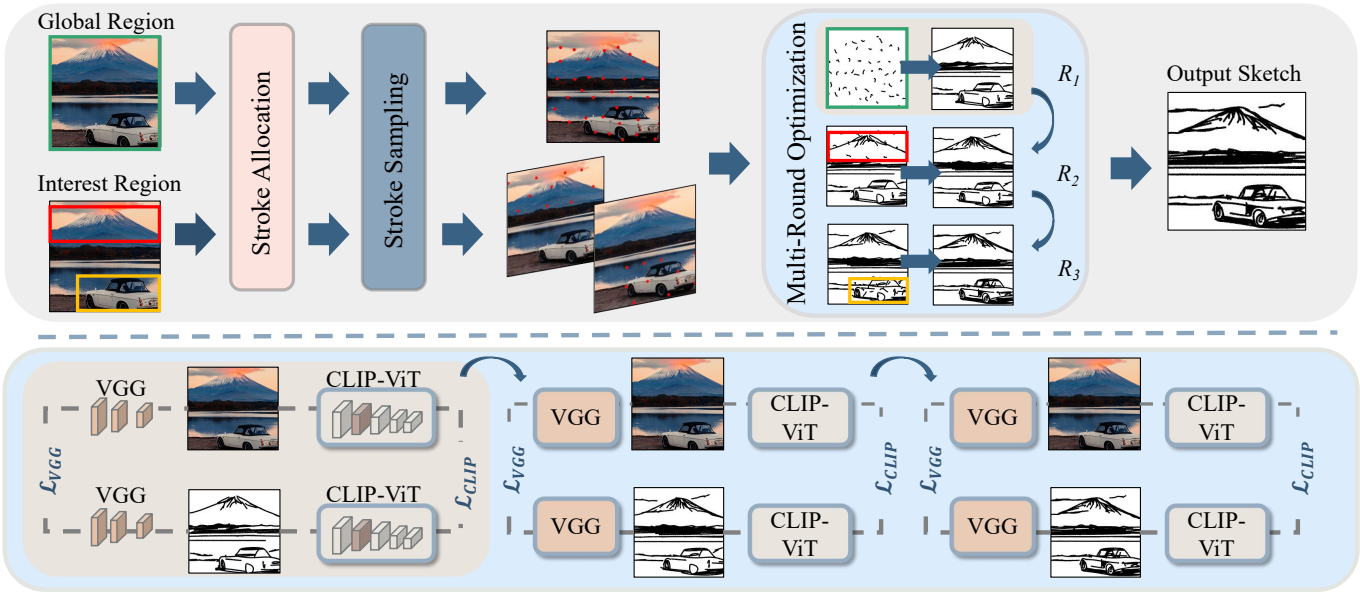

Fig. 2. Method overview – Given a scene photograph, and the selected regions, stroke initialization (stroke allocation and stroke sampling) is used to determine the initial strokes locations (red points) of different regions. These initial stroke locations will be converted into Bézier curves (black curves) as input of our multi-round optimization. In the bottom we show the details of the loss function during optimization.

the scene's semantics. We introduce a VGG16 [13] model to promote the visual consistency and similarity between a sketch and an image.

We evaluate our proposed method for various photographs, including people, nature, indoor, animals et al., to showcase the effectiveness of our method. In summary, our contributions encompass four aspects: (1) We propose a novel method to convert a scene photograph into a sketch by optimizing it in a region-by-region fashion. A variety of sketching results can be achieved by adjusting regions in our scheme. (2) We apply a farthest point sampling (FPS) algorithm to the input image, evenly sampling positions on region edges as stroke positions, which are wisely utilized to emphasize content information. (3) We introduce two novel loss functions, a CLIP-based Semantic Loss and a VGG-based Feature Loss. These losses improve the generation of sketches, reflecting a balanced combination of both semantic and geometric features.

II. Method

We present a new method to processively convert a given scene photograph into a sketch with multiple rounds of optimization. An overview of our method can be seen in Fig. 2. Briefly, given an arbitrary image, our method can recursively learn its different regions by adding optimiziable Bézier curves. We define our sketch as some sets of black Bézier curves from different regions placed on a white background. Firstly, we introduce a stroke allocation method to reasonably divide the total number of strokes into different regions. Then we use a proposed stroke sampling to determine stroke locations, which will be converted into our initial strokes (Bézier curves). To improve convergence, we define the order from the global region to other regions, optimizing strokes successively.

A. Stroke Initialization

Stroke Allocation. Our method is based on multiple rounds of stroke superposition. Thus we first should consider the allocation of strokes in different regions. As a case of fairness, we allocate strokes according to the edge points in each region by, based on the edge information gathered from contents of regions by edge detection [17]. $EdgeDetector$ in the following formula represents the edge extractor to gain the number of edge points in the region R_i. r presents the number of regions. The stroke allocation ratio of region R_i is calculated as follows:

$$E_i = EdgeDetector(R_i) \tag{1}$$

$$N_{E_i} = len(E_i) \tag{2}$$

$$Ratio_i = \frac{N_{E_i}}{\sum_i^r N_{E_i}} \tag{3}$$

N_s presents the number of total strokes. We then compute the final stroke allocation of region i:

$$N_i = Ratio_i \cdot N_s \tag{4}$$

Stroke Sampling. As each stroke corresponds to a sample point, the distribution of those points determines the distribution of strokes. For uniform coverage of image information and to reduce the lack of information, we adopt a farthest point sampling (FPS) [18] (the process shown in Fig. 3). As a first step, we process the region to get the edges. Next, we use a farthest point sampling to sample strokes, selecting a subset of points from a larger set and maximizing the minimum distance between any two selected points. The process ensures that the selected points are well-spaced and representative of entire set.

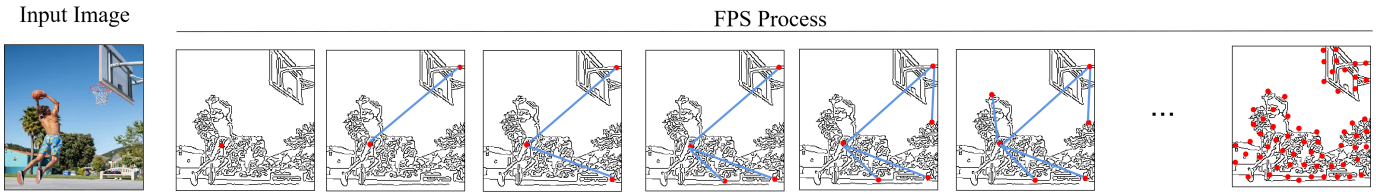

Fig. 3. An illustration of FPS process, which ensures that the sample points are well-spaced in the edge image.

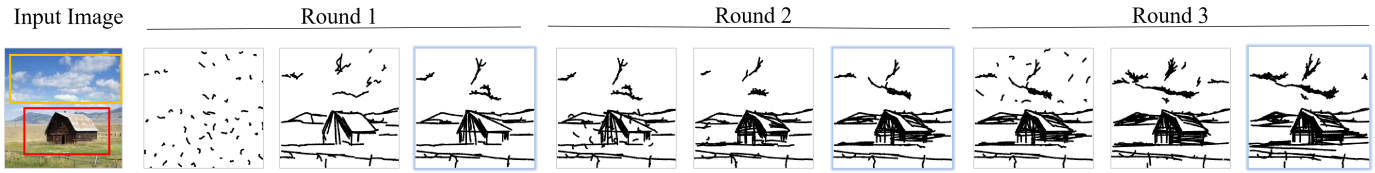

Fig. 4. The process of the optimization. The optimization results of the previous round will be superimposed with the initial strokes of the current region as the optimization input for the next round.

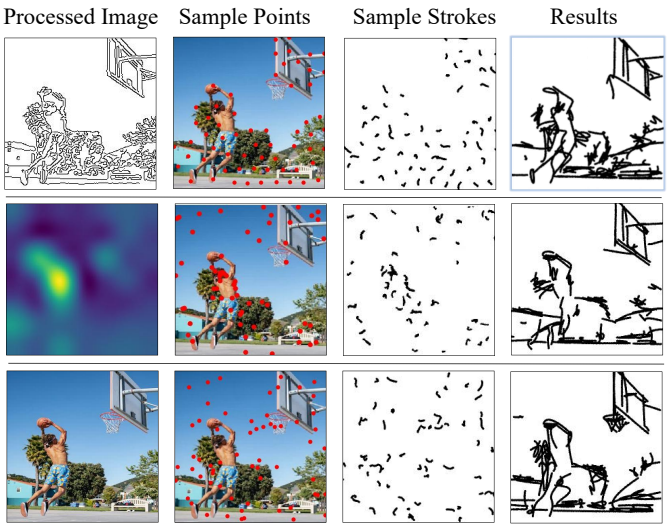

Fig. 5. Different Stroke Sampling Method. Top to Bottom: FPS sampling method, the sampling method of CLIPasso [8], random sampling.

In Formula 5, we get a stroke sampling point set corresponding to the number of strokes in the region R_i.

$$\{p_1, p_2, ..., p_n\} = EdgeDetector(R_i) \tag{5}$$

$$\{p_{k_1}, p_{k_2}, ..., p_{k_{N_i}}\} = FPS(\{p_1, p_2, ..., p_n\}, N_i) \tag{6}$$

Fig. 5 shows that our stroke sampling method contributes significantly to the quality of the final sketch compared to the sampling method of CLIPasso [8] and random sampling. We further analyze the effect and variability of FPS in the supplementary material.

B. Loss Function

In previous works, some commonly used loss functions to minimize the error between images and results based on pixel-wise metrics. Although pixel-wise loss is simple yet intuitive, it is not sufficient to measure the distance between sketches and images as a sketch is highly sparse and abstract. To address this, we leverage a pre-trained CLIP model to guide the training process.

CLIP-based Semantic Loss. For encoding shared information from both sketches and images, we follow CLIPasso [8] using CLIP model to compute the distance between the embeddings of the sketch $CLIP(Sketch)$ and image $CLIP(Image)$ as:

$$\mathcal{L}_{CLIP1} = dist(CLIP(Sketch), CLIP(Image)) \tag{7}$$

where $dist(x, y) = 1 - \frac{x \cdot y}{\|x\| \cdot \|y\|}$ is the cosine distance. And we are also based on the definition of the L2 distance loss function between certain activation layer l in the CLIP model:

$$\mathcal{L}_{CLIP2} = \|CLIP_l(Sketch) - CLIP_l(Image)\|_2^2 \tag{8}$$

with $\lambda = 0.1$, the CLIP-based Semantic Loss of the optimization is then defined as:

$$\mathcal{L}_{CLIP} = \mathcal{L}_{CLIP1} + \lambda \mathcal{L}_{CLIP2} \tag{9}$$

VGG-based Feature Loss. To further improve the structural similarity between the image and the sketch, we compute the L2 distance between the feature of them based on VGG16 [13]:

$$\mathcal{L}_{VGG} = \|VGG(Sketch) - VGG(Image)\|_2^2 \tag{10}$$

More details about the CLIP model, activation layer and VGG model could be find in the supplementary material. The final objective of the optimization is then defined as:

$$\mathcal{L}_{SUM} = \mathcal{L}_{CLIP} + \mathcal{L}_{VGG} \tag{11}$$

C. Optimization

In each optimization round, stroke parameters are overlaid with those from the previous round, with all parameters trained for 800 iterations. We start with a low-level sketch from the global region in the first round of optimization, refining it by incorporating strokes from other user-selected regions. As strokes from subsequent regions are sampled, each optimization round yields a flexible sketch. Fig. 4 illustrates this process, showcasing sketches at varying levels of abstraction across different optimization rounds. It should be noted that multiple rounds, while allowing for careful optimization of regions, can also lead to excessive overlap between regions, affecting the final result.

TABLE I

COMPARISON OF THE LPIPS [19] SCORE AND SSIM [20] SCORE. THE SCORE FROM LEFT TO RIGHT CORRESPOND TO THE RESULTS OF OUR METHOD, CLIPASSO [8] AND CLIPASCENE [14] BASED ON SKETCH RESULTS AT LOW AND HIGH ABSTRACTION LEVELS.

	Ours		CLIPasso		CLIPascene		XoG	Photo-Sketching	Chan et al.	UPDG
LPIPS ↓	0.566	0.519	0.672	0.602	0.577	0.525	0.549	0.728	0.520	0.640
SSIM ↑	0.654	0.690	0.613	0.622	0.618	0.676	0.740	0.334	0.652	0.649

TABLE II

COMPARISON OF USER PREFERENCE RATES. THE SCORES FROM LEFT TO RIGHT CORRESPOND TO THE RESULTS OF OUR METHOD, CLIPASSO [8] AND CLIPASCENE [14] ARE BASED ON SKETCHING RESULTS AT LOW AND HIGH ABSTRACTION LEVELS.

	Ours v.s. CLIPasso		Ours v.s. CLIPascene		Ours v.s. XoG	Ours v.s. Photo-Sketching	Ours v.s. Chan et al.	Ours v.s. UPDG
Ours	88.7%	96.3%	12.5%	38.6%	41.9%	96.2%	42.5%	40.0%
Others	5.0%	2.5%	10.0%	26.4%	38.4%	3.8%	32.5%	28.8%
Equal	6.3%	1.2%	77.5%	60.0%	19.7%	0.0%	25.0%	32.2%

III. EXPERIMENTS AND RESULTS

A. Implementation

Optimization Details. We use Adam optimizer with a learning rate set to 1. We evaluate the output sketch every iterations. Evaluation is done by computing the loss without random augmentations. We repeat the optimization process until convergence (when the difference in loss between two successive evaluations is less than 0.00001), this typically takes around 800 iterations. It takes 4 minutes to run 800 iterations on our server NVIDIA RTX 3090 GB, however, after 500 iterations, it is already possible to get a recognizable sketch for most images.

Curves Details. For each curves, we sample positions for their first control points using the FPS algorithm and then randomly sample the next three control points of each Bezier curve within a small radius (0.05) of the first point.

Edge Detection. We use the Canny edge detection [17] on the preprocessed image. It takes the grayscale image as input along with two threshold values: the lower threshold and the upper threshold. We determining the appropriate threshold values as (20, 200) by a trial-and-error process.

B. Evaluation Metrics

We calculate the Learned Perceptual Image Patch Similarity (LPIPS) [19] and Structural Similarity Index (SSIM) [20] to evaluate models. The LPIPS assesses perceptual similarity by computing the L2 distance between deep feature representations extracted from pretrained convolutional neural networks, such as VGG [13]. It measures the perceptual quality and semantic similarity between the synthetic images and the ground-truth images. The SSIM evaluates structural similarity by analyzing luminance, contrast, and structure between image pairs, providing a value between 0 and 1, where 1 indicates identical images.

C. Comparison to State-of-the-art Methods

We conduct a comparison of our method against various alternative sketching techniques, such as XoG [21], Photo-Sketching [22], Chan et al. [23], and UPDG [24]. It is worth noting that none of these techniques possess the capability to adjust the abstraction level of the sketches or produce vector-based sketches. We also provide comparisons with vector-based methods CLIPasso [8] and CLIPascene [14]. Considering that the sketches generated by XoG [21], Photo-Sketching [22], Chan et al. [23] and UPDG [24] are at different levels of abstraction, for a more comprehensive comparison, we use two different numbers of strokes (128 and 32) as inputs to our method, CLIPasso [8] and CLIPascene [14] to generate sketches at different levels of abstraction. Quantitative results in Table I show our mehod achieves competitive LPIPS and SSIM scores compared to existing methods, indicating high perceptual similarity and structural consistency. While XoG achieves slightly better scores in SSIM (0.740) and LPIPS (0.549), its inability to control abstraction levels limits its flexibility compared to our method.

D. User Study and Results

User study. To evaluate how well the sketches depict the input scene, we conducted a user study. We use 30 scene images to compare our sketches with six methods: CLIPasso [8], CLIPascene [14], XoG [21], Photo-Sketching [22], Chan et al. [23] and UPDG [24]. The participants were presented with the input image along with two sketches, one produced by our method and the other by the alternative method. In order to make a fair study, we compared Photo-Sketching with our sketches at the lower abstraction level. Table II presents the final preference rates, showing MROSS gains higher acceptance from participants.

Results. Fig. 6 and Fig. 7 demonstrate comparisons of MROSS with prior state-of-the-art methods. In Fig. 6 we select XoG [21], Photo-Sketching [22], Chan et al. [23], and UPDG [24], which do not have the ability to adjust the sketch abstraction level or generate vector-based sketches, for comparison. Given the different levels of abstraction of these methods, we choose to demonstrate the different abstraction level generation of our method by generating three different number of strokes (128, 64 and 32). In Fig. 7, we provide comparisons with vector-based methods CLIPasso [8] and CLIPascene [14]. Both can generate sketches at different abstraction levels and generate sketches in vector format. Therefore, we set the number of strokes at different abstraction levels (128 and 32) among these two methods for better comparison.

In contrast, MROSS, benefiting from alignment in structure and sematics, is able to maintain recognizability, underlying structure, and essential visual components of the scene drawn

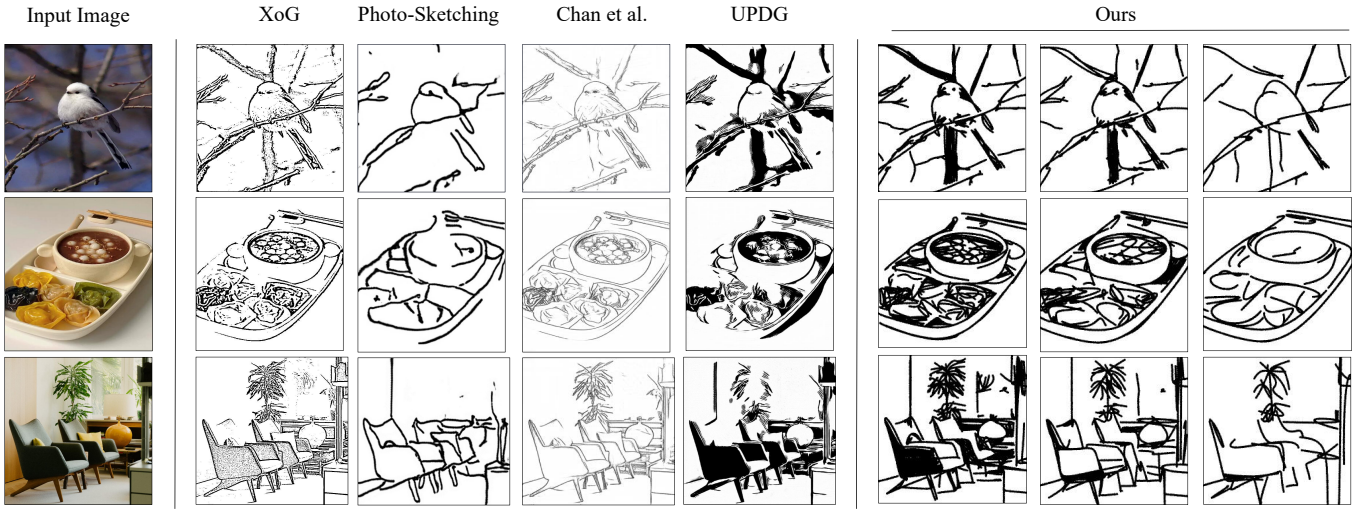

Fig. 6. Comparisons to methods that generate pixel-based sketches. On the right, are three representative sketches produced by our method depicting three levels of abstraction. More comparisons are provided in the supplementary material. More comparisons are provided in the supplementary material.

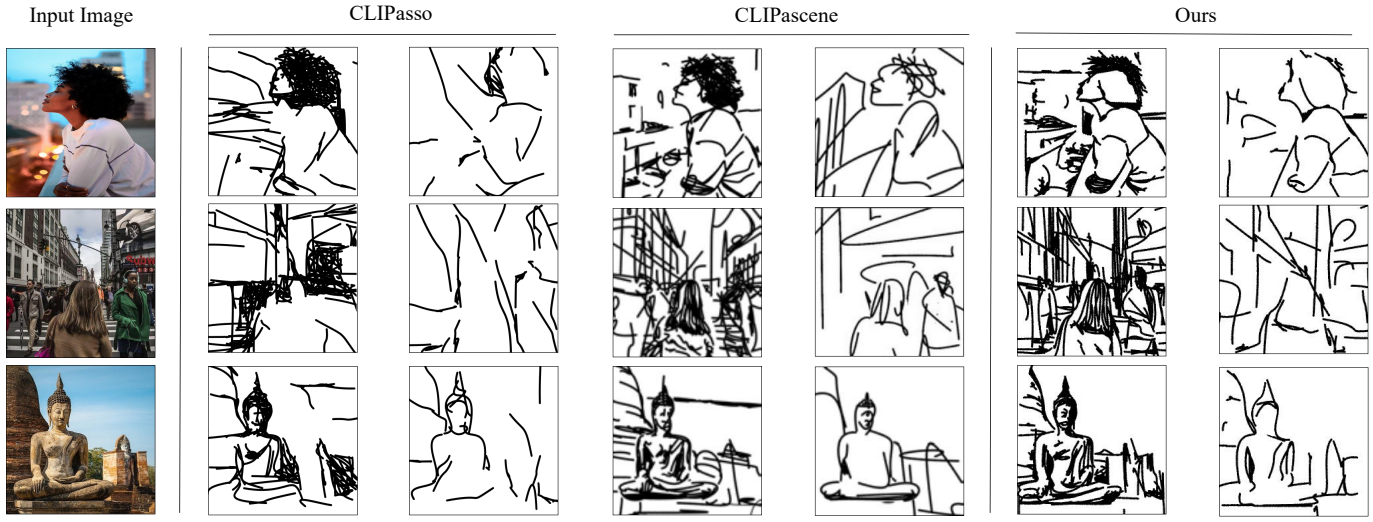

Fig. 7. Comparisons to methods that generate vector-based sketches. Representative sketches at high and low level of abstraction are generated.

regardless of the level of abstraction. CLIPasso [8] aims to portray objects accurately. However, for the scene, it fails to depict the total image in details. Then we compared to CLIPascene [14], for a fair comparison, we use the same decomposition technique to separate the input images into foreground and background. Then we use our method to sketch each part separately before combining them. CLIPascene [14] captures the whole image, which prevents them from emphasizing important regions of the input image. As shown in Fig. 7, our method is able to capture the entire image while displaying contours and details of key objects.

E. Ablation Study

In Fig. 8 we present a comprehensive analysis of the results obtained by the results of various loss functions. Fig. 8 clearly shows that CLIP-based Semantic Loss can robustly capture semantic concepts but falls short in terms of structural fidelity. VGG-based Feature Loss could perfectly preserve the geometric structure, ensuring a faithful representation of the original form. Ablating these loss functions can prove that our final loss combination not only achieves semantic accuracy but also effectively preserves complex geometric details, achieving a harmonious balance between semantic expression and structural integrity.

IV. CONCLUSION

We propose MROSS, an innovative scene sketching method that adopts a multi-round optimization strategy focusing on autonomously selected different regions. This enables a coarse-to-concrete representation, which is particularly beneficial for processing complex images. Meanwhile, we guide sketch generation with CLIP-based Semantic Loss and VGG-based Feature Loss. Extensive experiments and user studies confirm the superior scene generation performance of MROSS compared to existing methods.

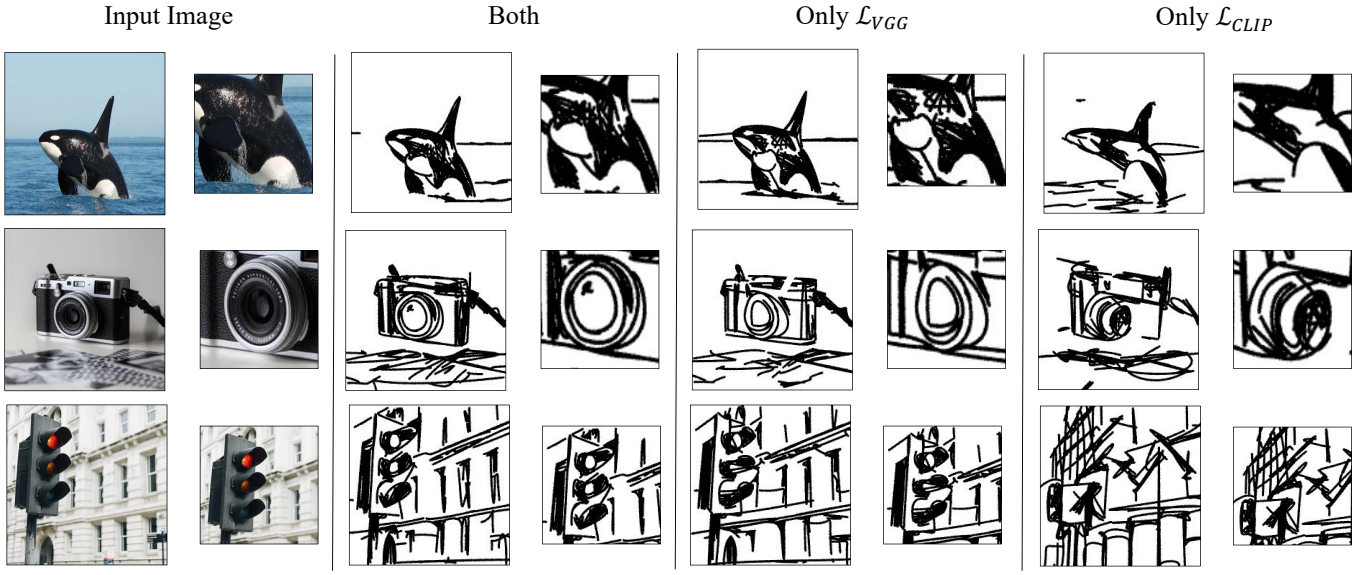

Fig. 8. Ablation of loss function. The effect of using only the CLIP-based Semantic Loss: capture abstract semantic details, like background, light and shadow. The effect of using only the VGG-based Feature Loss: emphasize structural features.

V. ACKNOWLEDGEMENT

This research was supported by National Natural Science Foundation of China (Grant No., 62202152 & 62202151).

REFERENCES

[1] Zhenbei Wu, Haoge Deng, Qiang Wang, Di Kong, et al., "Sketchscene: Scene sketch to image generation with diffusion models," in *2023 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2023, pp. 2087–2092.

[2] Subhadeep Koley, Ayan Kumar Bhunia, Aneeshan Sain, et al., "Picture that sketch: Photorealistic image generation from abstract sketches," in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2023, pp. 6850–6861.

[3] Feng-Lin Liu, Hongbo Fu, Yu-Kun Lai, and Lin Gao, "Sketchdream: Sketch-based text-to-3d generation and editing," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 4, pp. 1–13, 2024.

[4] Samaneh Azadi, Matthew Fisher, Vladimir G Kim, Zhaowen Wang, Eli Shechtman, and Trevor Darrell, "Multi-content gan for few-shot font style transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7564–7573.

[5] Yang Chen, Yu-Kun Lai, and Yong-Jin Liu, "Cartoongan: Generative adversarial networks for photo cartoonization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9465–9474.

[6] Phillip Isola, Jun-Yan Zhu, et al., "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[7] Zhichao Liao, Fengyuan Piao, Di Huang, et al., "Freehand sketch generation from mechanical components," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 6755–6764.

[8] Yael Vinker, Ehsan Pajouheshgar, Jessica Y Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir, "Clipasso: Semantically-aware object sketching," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–11, 2022.

[9] Itamar Berger, Ariel Shamir, Moshe Mahler, Elizabeth Carter, and Jessica Hodgins, "Style and abstraction in portrait sketching," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, pp. 1–12, 2013.

[10] Mengsi Guo, Mingfu Xiong, Jin Huang, et al., "Face photo-sketch portraits transformation via generation pipeline," *The Visual Computer*, pp. 1–14, 2024.

[11] Alec Radford, Jong Wook Kim, et al., "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[13] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] Yael Vinker, Yuval Alaluf, Daniel Cohen-Or, and Ariel Shamir, "Clipascene: Scene sketching with different types and levels of abstraction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4146–4156.

[15] Kevin Frans, Lisa Soros, et al., "Clipdraw: Exploring text-to-drawing synthesis through language-image encoders," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5207–5218, 2022.

[16] Peter Schaldenbrand, Zhixuan Liu, and Jean Oh, "Styleclipdraw: Coupling content and style in text-to-drawing translation," *arXiv preprint arXiv:2202.12362*, 2022.

[17] John Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, , no. 6, pp. 679–698, 1986.

[18] Charles Ruizhongtai Qi, Li Yi, et al., "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[19] Richard Zhang, Phillip Isola, Alexei A Efros, et al., "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[20] Zhou Wang, Alan C Bovik, et al., "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[21] Holger Winnemöller, Jan Eric Kyprianidis, et al., "Xdog: An extended difference-of-gaussians compendium including advanced image stylization," *Computers & Graphics*, vol. 36, no. 6, pp. 740–753, 2012.

[22] Mengtian Li, Zhe Lin, Radomir Mech, et al., "Photo-sketching: Inferring contour drawings from images," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 1403–1412.

[23] Caroline Chan, Frédo Durand, and Phillip Isola, "Learning to generate line drawings that convey geometry and semantics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7915–7925.

[24] Ran Yi, Yong-Jin Liu, Yu-Kun Lai, and Paul L Rosin, "Unpaired portrait drawing generation via asymmetric cycle mapping," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8217–8225.