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

Yiqi Liang, Ying Liu, Dandan Long, Ruihui Li[*]

College of Computer Science and Electronic Engineering, Hunan University, China
{yiqiliang, liu_ying, ddlong, 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 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. We optimize the different regions of input scene in multiple rounds. In each round of optimization, strokes sampled from the next region can seamlessly be integrated into the sketch generated in the previous round of optimization. We propose 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.

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 the 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.

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. (Azadi et al. 2018; Chen, Lai, and Liu 2018; Isola et al. 2017; Li and Wand 2016) 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.

---

[*]Corresponding author.

Besides, different regions within a scene may have varying levels of importance or prominence. 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 (Muhammad et al. 2018; Vinker et al. 2022b) or portrait sketching (Berger et al. 2013), and often simply use the number of strokes to define the effect of their sketches.

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 (Radford et al. 2021; Dosovitskiy et al. 2020) and VGG16 model (Simonyan and Zisserman 2014).

Unlike (Frans, Soros, and Witkowski 2022; Schaldenbrand, Liu, and Oh 2022), optimization is performed for different regions in our method, which helps highlight regions of interest, achieving coarse-to-concrete sketches. An intuitive and succinct learning process can be seen in Figure 1). 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. Figure 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 (Dosovitskiy et al. 2020) to guide the optimization, where encourages the creation of looser sketches that emphasize the scene's semantics. We introduce a VGG16 (Simonyan and Zisserman 2014) 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. Our main contributions in this work can be summarized as follows:

- We propose a novel method to convert a scene photograph into a sketch by optimizing it in a region-by-region
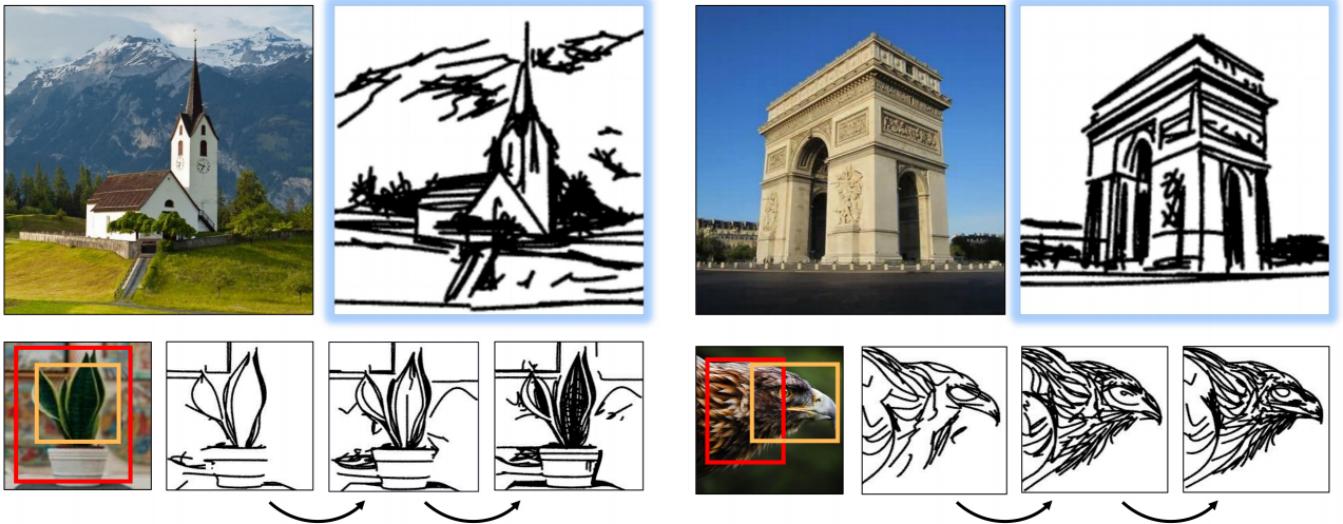
Figure 1: Our method converts a scene image into a sketch with different levels of abstraction. The first row presents a concise, comprehensive vector sketch of the scene, ideal for designer edits. The second row illustrates the multi-round optimization process, achieving coarse-to-concrete sketches without altering the total stroke count.

fashion. A variety of sketching results can be achieved by adjusting regions in our scheme.

- 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.

- We introduce two novel loss functions, a CLIP-Based Semantic loss and a VGG-Based Feature loss. These losses improve sketch generation, reflecting a balanced combination of both semantic and geometric features.

Related Work

Photo-Sketch Synthesis. Traditional photo-sketch synthesis methods are mainly based on image processing techniques, such as convolution, filtering, and grayscale, etc., to obtain corresponding sketches style pictures. However, the generated sketch of these methods are of low quality. Researchers investigated approaches that related to the deep learning, considering photo-sketch as a cross domain image-to-image translation problem. (Muhammad et al. 2018) trained a model for abstract sketch generation through reinforcement learning of a stroke removal policy that learns to predict which strokes can be safely removed without affecting recognizability. (Li et al. 2019) proposed a learning-based method to resolve the diversity in the annotation of datasets. Pinz (Kampelmuhler and Pinz 2020) employed a fully convolutional encoder-decoder structure to accomplish a mapping from image space to sketch space.

Yet, recently, CLIPasso (Vinker et al. 2022b) proposed a novel object sketching method that can achieve different levels of abstraction, guided by geometric and semantic simplifications. They defined a sketch as a set of Bézier curves and extracted the salient regions of the input image to define the initial locations of the strokes. In contrast to CLIPasso

(Vinker et al. 2022b), our method is not restricted to objects and can handle the challenging task like scene sketching. Additionally, while they only utilized a single form of optimization, we disentangle optimization into multiple rounds controlling the fidelity of different region in the input image, which allows for a wider range of editing and manipulation.

Vector Graphics. Vector representations are widely used for a variety of sketching tasks and applications, combining with a number of deep learning models including RNN (Ha and Eck 2017), CNNs (Chen et al. 2017), BERT (Lin et al. 2020), Transformers (Bhunia et al. 2020; Ribeiro et al. 2020), GANs (Balasubramanian, Balasubramanian et al. 2019) and reinforcement learning algorithms (Ganin et al. 2018; Mellor et al. 2019; Zhou et al. 2018). While traditional image generation methods operating over vector images require a vector-based dataset, (Li et al. 2020; Mihai and Hare 2021) had shown its possible to manipulate or synthesize vector content by using raster-based loss functions bypassing this limitation (that is, the process of actually drawing the vectors into an image is not part of the learning machinery). Among them, we consider to use the method of (Li et al. 2020), as it can flexibly handle a wide range of curves and strokes, including Bézier curves.

CLIP-Based Image Vectorization. (Radford et al. 2021) proposed CLIP, which is a neural network trained on 400 million image-text pairs collected from the internet with the objective of creating a joint latent space. Being trained on a wide variety of image domains along with lingual concepts, CLIP models are found to be very useful for a wide range of zero-shot tasks, and have enabled a number of successful methods for drawings, such as CLIPDraw (Frans, Soros, and Witkowski 2022), StyleCLIPDraw (Schaldenbrand, Liu, and Oh 2022), CLIP-CLOP (Mirowski et al. 2022), and CliPascene (Vinker et al. 2022a). (Tian and Ha 2022) employed evolutionary algorithms combined with CLIP, to produce
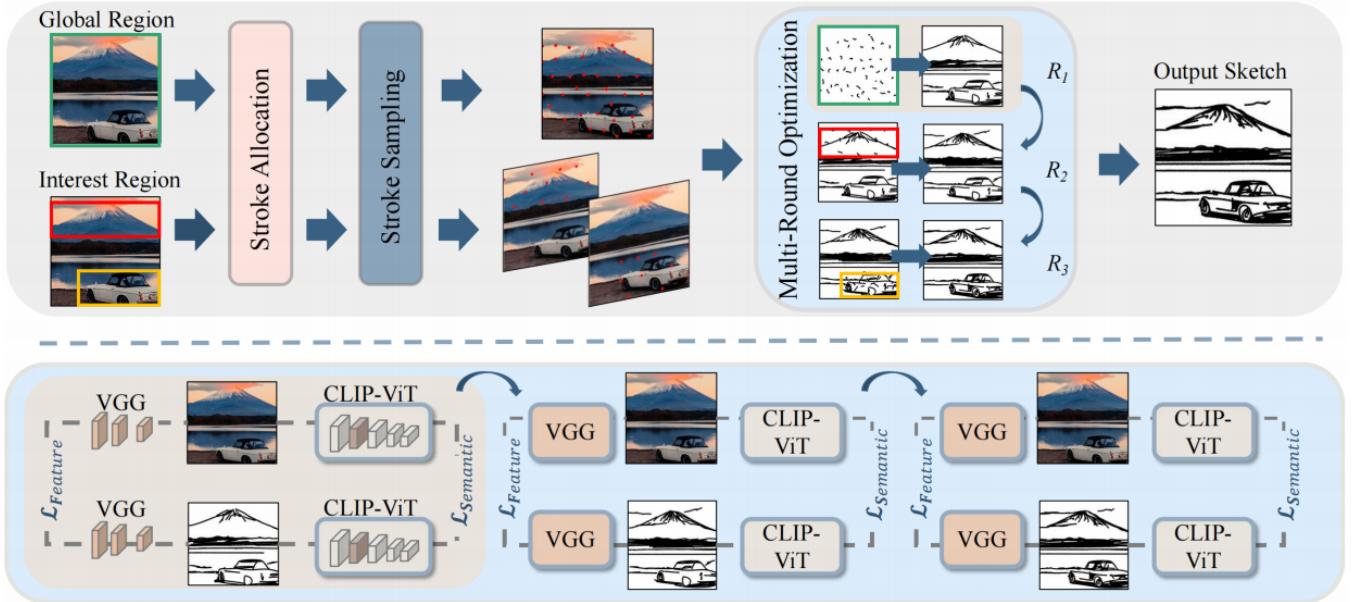
Figure 2: Method overview – Given a scene photograph, and the selected regions, stroke initialization (stroke allocation and stroke sampling) is used to determine the initial strokes locations (red points) of different regions. These initial stroke locations will be converted into Bézier curves (black curves) as input of our multi-round optimization. In the bottom we show the details of the loss function during optimization.

creative abstract concepts represented by colored triangles guided by text or shape. Their results are limited to either fully semantic (using CLIP's text encoder) or entirely geometric (using L2). CLIPasso (Vinker et al. 2022b) introduced a new loss term and a saliency-guided initialization procedure. Our method also utilizes the CLIP image encoder to guide the process of converting a photograph to a sparse sketch with a combination of features that capture the essence of the input image.

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 Figure 2. Briefly, given an arbitrary image, our method can recursively learn its different regions by adding optimizable 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 to optimize strokes successively.

There are several advantages behind our method. First, it considers the information from different regions of the input image separately, which can help ignore background interference (see Figure 7). Also, the sketch results at the same level of abstraction is greatly increased based on different choices of the regions (see Figure 8). Last, during our opti-

mization process, we also can easily obtain sketches results at different levels of abstraction without changing the total number of strokes (see Figure 1).

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, based on the information gathered from each region's edge contents. $EdgeDetector$ 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) (the process shown in Figure 4). As a first step, we process the region to get the edges. Next, we use a farthest point sampling to sample strokes, selecting
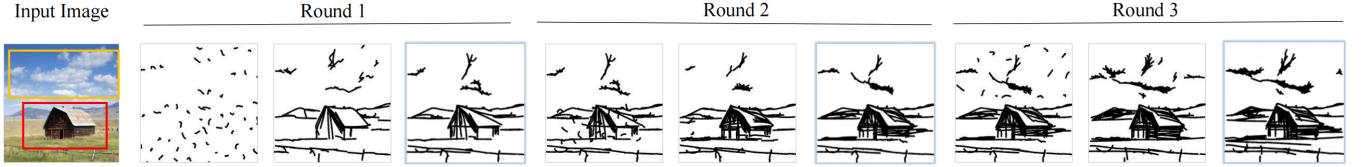
Figure 3: The process of the optimization. The optimization results of the previous round will be superimposed with the initial strokes of the current region as the optimization input for the next round.
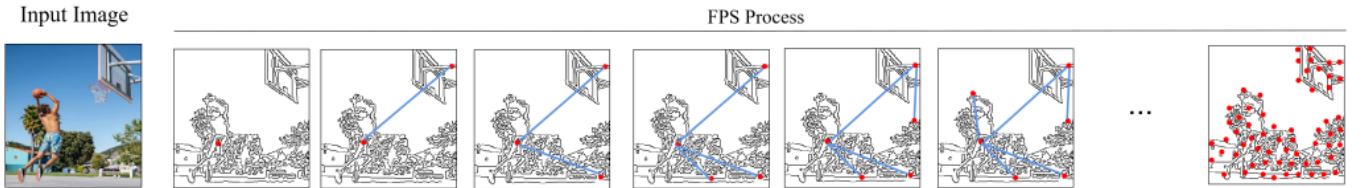

Figure 4: An illustration of FPS process, which ensures that the sample points are well-spaced in the edge image.
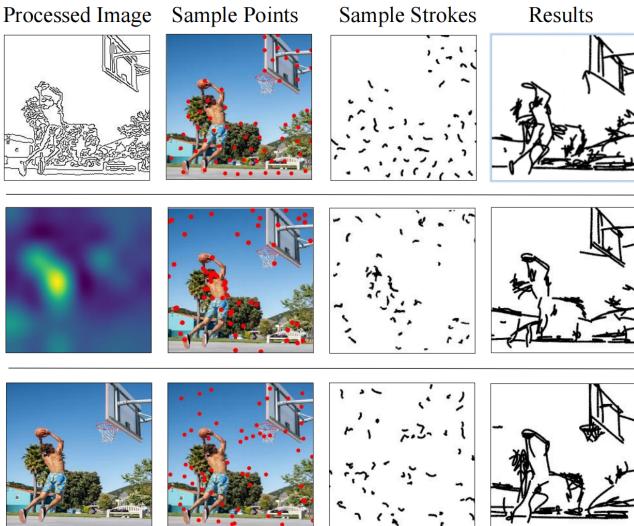

Figure 5: Different Stroke Sampling Method. Top to Bottom: our proposed sampling method (FPS), the sampling method of CLIPasso (Vinker et al. 2022b), random sampling.

a subset of points from a larger set and maximizing the minimum distance between any two selected points. This process ensures that the selected points are well-spaced and representative of the entire set.

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

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

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

Figure 5 shows that our stroke sampling method contributes significantly to the quality of the final sketch compared to the sampling method of CLIPasso (Vinker et al. 2022b) and random sampling. We further analyze the effect and variability of FPS in the supplementary material.

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. Due to the capabilities of encoding shared information from both sketches and images, we follow the work of (Vinker et al. 2022b) using CLIP model to compute the distance between the embeddings of the sketch $CLIP(Sketch)$ and image $CLIP(Image)$ as:

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

where $dist(x, y) = 1 - \frac{x \cdot y}{\|x\| \cdot \|y\|}$ is the cosine distance. However, while (Vinker et al. 2022b) utilize the ResNet-based (He et al. 2016) CLIP model for sketching, we find that the ViT-based (Dosovitskiy et al. 2020) CLIP model provides better coverage of global context. Therefore the loss function is defined as the L2 distance between the activation of ViT-B/32 CLIP model at layers:

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

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

$$\mathcal{L}_{CLIP} = \mathcal{L}_{CLIP1} + \lambda \mathcal{L}_{CLIP2} \quad (9)$$

VGG-Based Feature Loss. To further improve the similarity between the image and the sketch, we compute the L2 distance between the feature of the sketch and the image based on VGG16 (Simonyan and Zisserman 2014):

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

For more details of VGG model we used please refer to the supplementary material. The final objective of the optimization is then defined as:

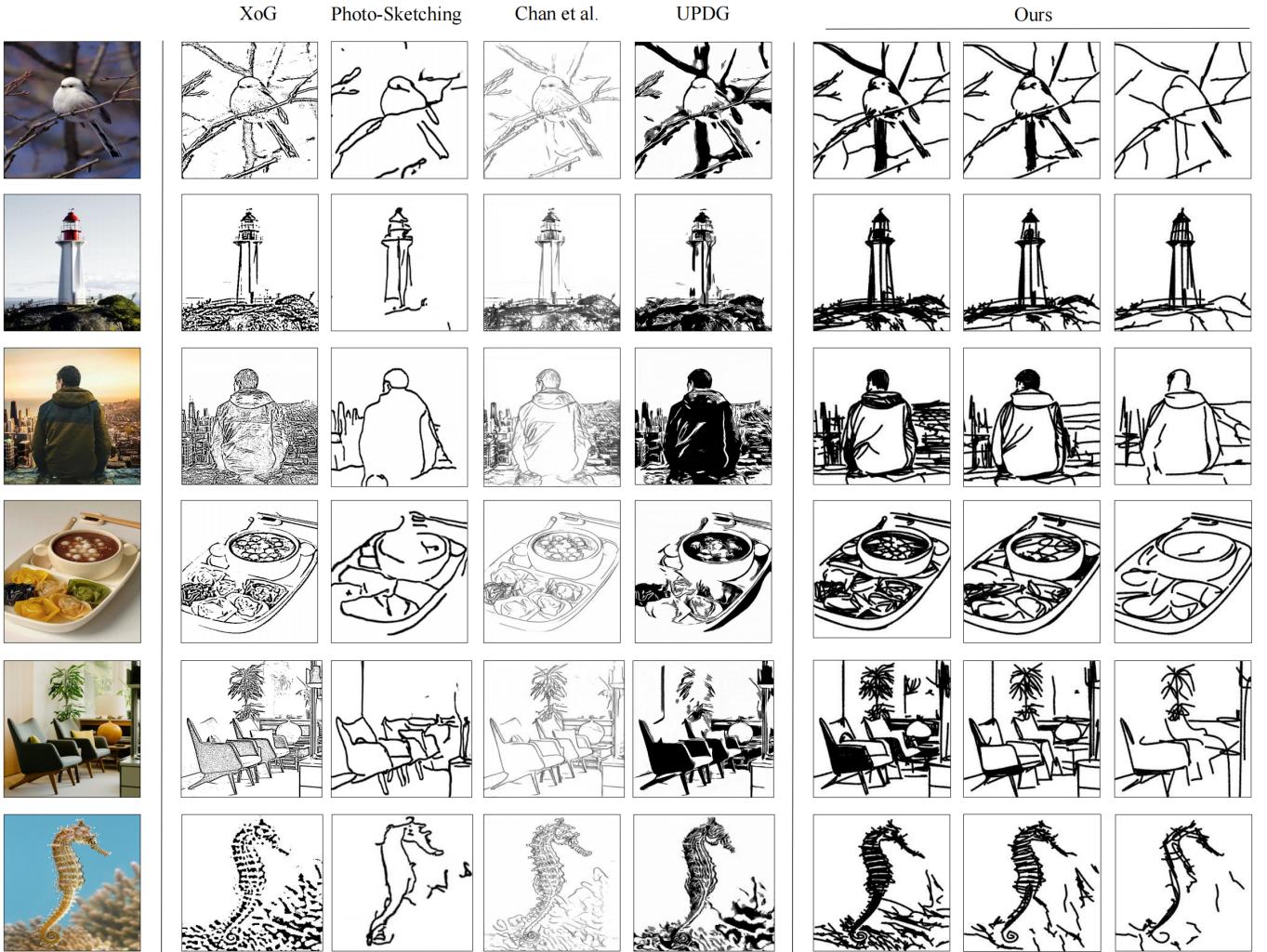$$\mathcal{L}_{SUM} = \mathcal{L}_{CLIP} + \mathcal{L}_{VGG} \quad (11)$$

Figure 6: The comparisons of scene sketching results. On the right, are three representative sketches produced by our method depicting three levels of abstraction. More comparisons are provided in the supplementary material.

Optimization

For each optimization round, the parameters of stroke are superimposed with the previous round. All parameters are trained for 800 iterations. We define that the first round of optimization is from the global region, and we are able to gain a sketch at a low level of abstraction. Then the sketch will be refined based on the superimposition of strokes from other regions selected by the user. Since the strokes superimposed are the results of stroke sampling from the next region, the sketch in each optimization is flexible.

In Figure 3, we show the process of our optimization and sketch results at different levels of abstraction generated by different optimization round.

Experiments

In the following, we demonstrate the performances of our technique qualitatively and quantitatively, and provide comparisons to state-of-the-art sketching methods. Figure 8 illustrates the impact of the regions. When different regions

are chosen, we may get some variations, with the tradeoff that it may ignore the other regions. Further analysis, results, and a user study are provided in the supplementary material.

Qualitative Evaluation

We compare our method with alternative sketching methods for different subjects, including XoG (Winnemöller, Kyprianidis, and Olsen 2012), Photo-Sketching (Li et al. 2019), Chan et al. (Chan, Durand, and Isola 2022) and UPDG (Yi et al. 2020). Note that, none of these sketching approaches can produce sketches with varying abstraction levels, and none can produce sketches in vector format. Due to the fact that our method requires a predefined number of strokes as input, we present three sketches produced by our method depicting three representative levels of abstraction.

The sketches produced by UPDG (Yi et al. 2020) and Chan et al. (Chan, Durand, and Isola 2022) are detailed, closely similar to the edge maps of the input images (like results of XoG (Winnemöller, Kyprianidis, and Olsen 2012)).

	Ours			XoG	CLIPasso	Photo-Sketching	Chan et al.	UPDG
LPIPS ↓	0.519	0.531	0.566	0.549	0.702	0.628	0.520	0.740
SSIM ↑	0.690	0.680	0.654	0.740	0.614	0.634	0.632	0.624

Table 1: Comparison of the LPIPS score and SSIM score. Our scores correspond from left to right to the sketches under the abstraction levels from left to right in Figure 6.
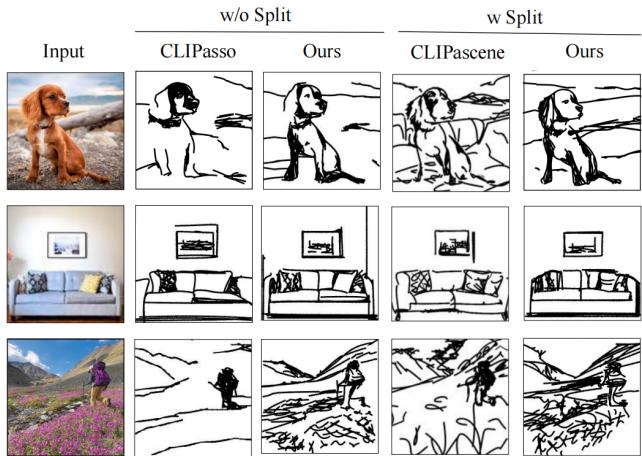

Figure 7: Comparison to CLIPasso (Vinker et al. 2022b) without scene decomposition and to CLIPascene (Vinker et al. 2022a) with scene decomposition.

The geometric structures of these sketches are most similar to the input images shown in the leftmost column of Figure 6. However, some edges and semantic information may be lacking. The sketches produced by Photo-Sketching (Li et al. 2019) are less detailed and difficult to align well with the input scene, which is difficult to identify the main content. Our generated sketches are able to maintain recognizability, underlying structure, and essential visual components of the scene drawn.

In Figure 7, we provide additional comparisons with CLIPasso (Vinker et al. 2022b) and CLIPascene (Vinker et al. 2022a). Both of them can produce sketches with varying abstraction levels and produce sketches in vector format. We generate sketches with the same number of strokes. CLIPasso (Vinker et al. 2022b) aims to portray objects accurately. However, for the scene, it fails to depict the total image in details. This drawback may result from its stroke sampling method based on an attention map.

Then we compared to CLIPascene (Vinker et al. 2022a), 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 (Vinker et al. 2022a) captures the whole image, which prevents them from emphasizing important regions of the input image. Our method is able to capture the entire image and display key regions controllably. For example, the dog in the first row.

Figure 8: Comparison of sketches obtained under different regions division at the same number of strokes.

Quantitative Evaluation

LPIPS. LPIPS (Zhang et al. 2018) measures the perceptual similarity between two images and lower LPIPS scores indicate less perceptual difference. We calculate the scores between input images and sketches of different methods. Table 1 shows the average LPIPS scores for each methods. Our results under high level of abstraction are close to those of Chan et al. (Chan, Durand, and Isola 2022), which means our sketches follow the content of the input image. The results of XoG (Winnemöller, Kyprianidis, and Olsen 2012) are not the best, it may because LPIPS does not only consider edge information.

SSIM. To measure the fidelity level of sketches, we compute the SSIM (Wang et al. 2004) score between each input image and the corresponding sketch. In Table 1, we show the average resulting scores, where a higher score indicates a higher similarity. The sketches by XoG (Winnemöller, Kyprianidis, and Olsen 2012) obtained high scores, which is consistent with our observation that SSIM highlight the edges of the image. Under different levels of abstraction, our method shows relatively good scores, indicating that we have a better effect on edge information preservation. This shows the effectiveness of our edge-based sampling method.

User Study. The user study examines how well the sketches depict the input scene. Since the absence of CLIPascene's (Vinker et al. 2022a) code, we use 30 scene images to compare our sketches with four methods: CLIPasso (Vinker et al. 2022b), Photo-Sketching (Li et al. 2019), Chan et al. (Chan, Durand, and Isola 2022) and UPDG (Yi et al. 2020).

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
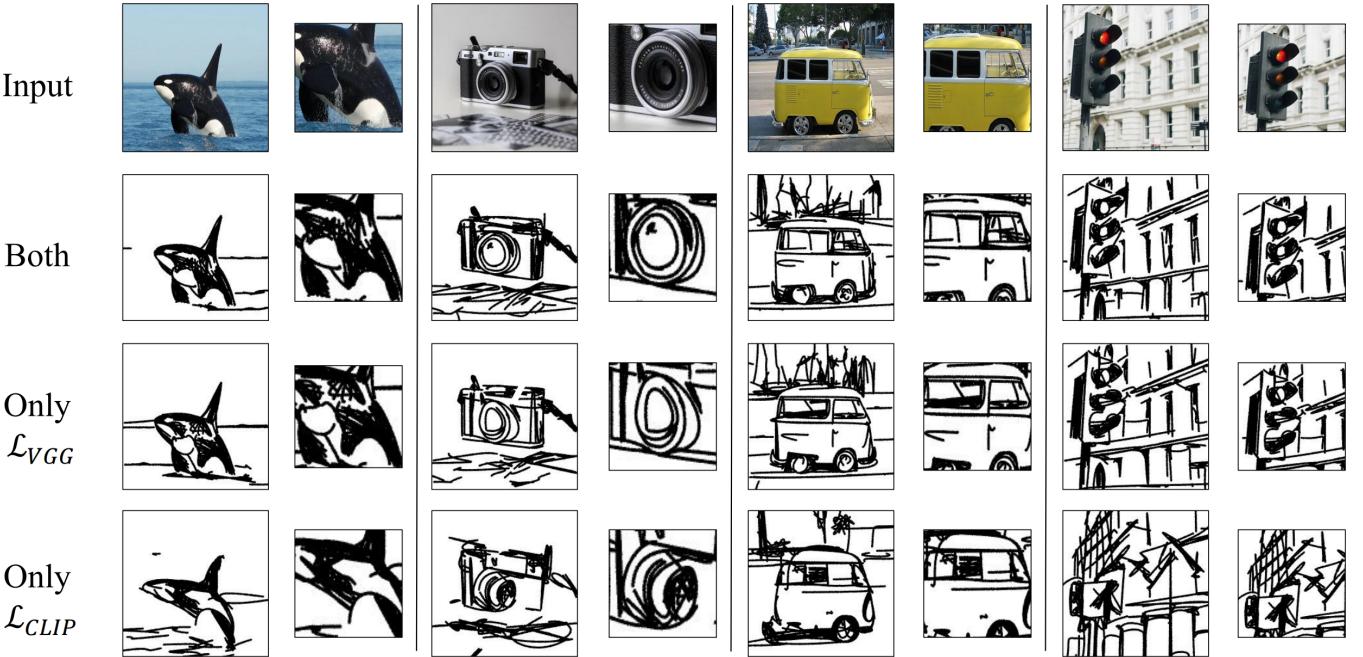
Figure 9: Comparison of different loss function. The effect of using only the CLIP-Based Semantic loss: Capture abstract semantic details, like background, light and shadow. The effect of using only the VGG-Based Feature loss: emphasize image features, such as geometric contours.

	Ours v.s. CLIPasso	Ours v.s. Photo-Sketching	Ours v.s. Chan et al.	Ours v.s. UPDG
Ours	93.0%	96.6%	42.7%	29.8%
Others	2.7%	3.4%	32.9%	24.1%
Equal	4.3%	0.0%	24.4%	46.1%

Table 2: Results of our user study. For each method, we specify the percent of responses that preferred our sketch, the sketch of the alternative method, or found the sketches to be similar in their ability to capture the scene semantics.

study, we compared Photo-Sketching (Li et al. 2019) with our sketches at the lower abstraction level. Conversely, we compared the sketches of Chan et al. (Chan, Durand, and Isola 2022) and UDPG (Yi et al. 2020), more detailed. Table 2 shows the average resulting scores among all participants. Compared to CLIPasso (Vinker et al. 2022b) and Photo-Sketching (Li et al. 2019), our method achieves significantly higher rates. As we can see, although sketches produced by Chan et al. (Chan, Durand, and Isola 2022) and UPDG (Yi et al. 2020) are highly detailed, 42.7% responses preferred our sketches when compared to Chan et al. (Chan, Durand, and Isola 2022) and 46.1% of responses considered our sketches is similar to UPDG's (Yi et al. 2020). The results of user study demonstrate that sketches produced by our method well represent the elements of the input scene.

Ablation Study

In Figure 9, we show the results of different loss function. As can be seen, in that case the VGG-Based Feature loss faithfully preserves geometric structure, while the CLIP-Based Semantic loss strongly conveys the semantic concept, however at the cost of structure. Using their combination can result in an ideal sketch, which is not only semantically accurate, but also retains geometric details.

Conclusion

We propose a multi-round optimization for scene sketching based on different regions. We progressively infers the sketch with the help of proposed stroke initialization and loss functions: a CLIP-Based Semantic loss and a VGG-Based Feature loss. We can investigate the coarse-to-concrete representation for complex images with different regions. Extensive experiments and human evaluation confirm the superior reconstruction performance of our method over the existing. In the emerging field of sketch generation, we hope our work will open the door for further research.

Acknowledgments

References

Azadi, S.; Fisher, M.; Kim, V. G.; Wang, Z.; Shechtman, E.; and Darrell, T. 2018. Multi-content gan for few-shot font style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7564–7573.

Balasubramanian, S.; Balasubramanian, V. N.; et al. 2019. Teaching gans to sketch in vector format. *arXiv preprint arXiv:1904.03620*.

Berger, I.; Shamir, A.; Mahler, M.; Carter, E.; and Hodgins, J. 2013. Style and abstraction in portrait sketching. *ACM Transactions on Graphics (TOG)*, 32(4): 1–12.

Bhunia, A. K.; Das, A.; Muhammad, U. R.; Yang, Y.; Hospedales, T. M.; Xiang, T.; Gryaditskaya, Y.; and Song, Y.-Z. 2020. Pixelor: A Competitive Sketching AI Agent. So you think you can sketch? *ACM Transactions on Graphics (TOG)*, 39(6): 1–15.

Chan, C.; Durand, F.; and Isola, P. 2022. Learning to generate line drawings that convey geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7915–7925.

Chen, Y.; Lai, Y.-K.; and Liu, Y.-J. 2018. Cartoongan: Generative adversarial networks for photo cartoonization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9465–9474.

Chen, Y.; Tu, S.; Yi, Y.; and Xu, L. 2017. Sketch-pix2seq: a model to generate sketches of multiple categories. *arXiv preprint arXiv:1709.04121*.

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

Frans, K.; Soros, L.; and Witkowski, O. 2022. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *Advances in Neural Information Processing Systems*, 35: 5207–5218.

Ganin, Y.; Kulkarni, T.; Babuschkin, I.; Eslami, S. A.; and Vinyals, O. 2018. Synthesizing programs for images using reinforced adversarial learning. In *International Conference on Machine Learning*, 1666–1675. PMLR.

Ha, D.; and Eck, D. 2017. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1125–1134.

Kampelmuhler, M.; and Pinz, A. 2020. Synthesizing human-like sketches from natural images using a conditional convolutional decoder. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 3203–3211.

Li, C.; and Wand, M. 2016. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2479–2486.

Li, M.; Lin, Z.; Mech, R.; Yumer, E.; and Ramanan, D. 2019. Photo-sketching: Inferring contour drawings from images. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1403–1412. IEEE.

Li, T.-M.; Lukáč, M.; Gharbi, M.; and Ragan-Kelley, J. 2020. Differentiable vector graphics rasterization for editing and learning. *ACM Transactions on Graphics (TOG)*, 39(6): 1–15.

Lin, H.; Fu, Y.; Xue, X.; and Jiang, Y.-G. 2020. Sketch-bert: Learning sketch bidirectional encoder representation from transformers by self-supervised learning of sketch gestalt. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6758–6767.

Mellor, J. F.; Park, E.; Ganin, Y.; Babuschkin, I.; Kulkarni, T.; Rosenbaum, D.; Ballard, A.; Weber, T.; Vinyals, O.; and Eslami, S. 2019. Unsupervised doodling and painting with improved spiral. *arXiv preprint arXiv:1910.01007*.

Mihai, D.; and Hare, J. 2021. Differentiable drawing and sketching. *arXiv preprint arXiv:2103.16194*.

Mirowski, P.; Banarse, D.; Malinowski, M.; Osindero, S.; and Fernando, C. 2022. Clip-clop: Clip-guided collage and photomontage. *arXiv preprint arXiv:2205.03146*.

Muhammad, U. R.; Yang, Y.; Song, Y.-Z.; Xiang, T.; and Hospedales, T. M. 2018. Learning deep sketch abstraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8014–8023.

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

Ribeiro, L. S. F.; Bui, T.; Collomosse, J.; and Ponti, M. 2020. Sketchformer: Transformer-based representation for sketched structure. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14153–14162.

Schaldenbrand, P.; Liu, Z.; and Oh, J. 2022. Styleclipdraw: Coupling content and style in text-to-drawing translation. *arXiv preprint arXiv:2202.12362*.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Tian, Y.; and Ha, D. 2022. Modern evolution strategies for creativity: Fitting concrete images and abstract concepts. In *International conference on computational intelligence in music, sound, art and design (part of evostar)*, 275–291. Springer.

Vinker, Y.; Alaluf, Y.; Cohen-Or, D.; and Shamir, A. 2022a. Clipascene: Scene sketching with different types and levels of abstraction. *arXiv preprint arXiv:2211.17256*.

Vinker, Y.; Pajouheshgar, E.; Bo, J. Y.; Bachmann, R. C.; Bermano, A. H.; Cohen-Or, D.; Zamir, A.; and Shamir,

A. 2022b. Clipasso: Semantically-aware object sketching. *ACM Transactions on Graphics (TOG)*, 41(4): 1–11.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.

Winnemöller, H.; Kyprianidis, J. E.; and Olsen, S. C. 2012. XDoG: An eXtended difference-of-Gaussians compendium including advanced image stylization. *Computers & Graphics*, 36(6): 740–753.

Yi, R.; Liu, Y.-J.; Lai, Y.-K.; and Rosin, P. L. 2020. Unpaired portrait drawing generation via asymmetric cycle mapping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8217–8225.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Zhou, T.; Fang, C.; Wang, Z.; Yang, J.; Kim, B.; Chen, Z.; Brandt, J.; and Terzopoulos, D. 2018. Learning to sketch with deep q networks and demonstrated strokes. *arXiv preprint arXiv:1810.05977*.