Instance Contour Adjustment via Structure-driven CNN

Shuchen Weng[1], Yi Wei[2], Ming-Ching Chang[3], and Boxin Shi*[1]

[1] NERCVT, School of Computer Science, Peking University
[2] Samsung Research America AI Center
[3] University at Albany, State university of New York
{shuchenweng,shiboxin}@pku.edu.cn
yi.wei1@samsung.com mchang2@albany.edu

Abstract. Instance contour adjustment is desirable in image editing, which allows the contour of an instance in a photo to be either dilated or eroded via user sketching. This imposes several requirements for a favorable method in order to generate meaningful textures while preserving clear user-desired contours. Due to the ignorance of these requirements, the off-the-shelf image editing methods herein are unsuited. Therefore, we propose a specialized two-stage method. The first stage extracts the structural cues from the input image, and completes the missing structural cues for the adjusted area. The second stage is a structure-driven CNN which generates image textures following the guidance of the completed structural cues. In the structure-driven CNN, we redesign the context sampling strategy of the convolution operation and attention mechanism such that they can estimate and rank the relevance of the contexts based on the structural cues, and sample the top-ranked contexts regardless of their distribution on the image plane. Thus, the meaningfulness of image textures with clear and user-desired contours are guaranteed by the structure-driven CNN. In addition, our method does not require any semantic label as input, which thus ensures its well generalization capability. We evaluate our method against several baselines adapted from the related tasks, and the experimental results demonstrate its effectiveness.

1 Introduction

A photo can be considered as a composition of a certain number of instance(s), and the contour of an instance separates itself from the other instances. By adjusting instance contours, a user can achieve superior photography experience which cannot be met with the real but fixed scenery and even photography professionalism. For example, in Fig. 1 where the lake and Merlion are two instances, users hope to dilate the contour of the lake shape to be a flying pigeon; or they hope to create fantastic scenes, *e.g.*, eroding the contour of the Merlion to remove its body. These are moments when an instance contour adjustment function can turn the table.

* Corresponding author.

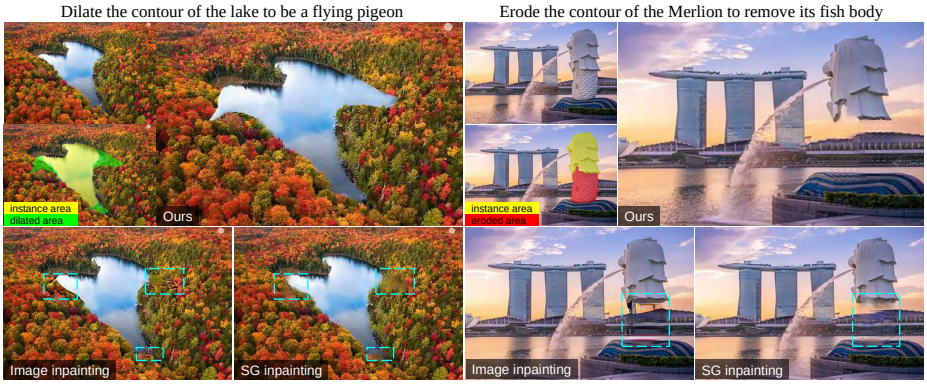


Fig. 1: **Task comparison.** The original and overlaid images are put on the left side of our results. The results of image inpainting [26] and semantic-guided (SG) inpainting [16] are shown below. The cyan dotted boxes highlight artifacts and the out-of-control contours.

In order to adjust the contour of an instance, users first extract the instance area (yellow in Fig. 1) with the image matting function. Then, users sketch the hypothetical contours which form two potential types of area: *(i)* **Dilated area** (refer to the left example) should be *exclusively* filled with the content of the instance area. *(ii)* **Eroded area** (refer to the right example) should be *exclusively* filled with the content external to the instance area.

These two *exclusion rules* distinguish the instance contour adjustment from the related tasks. In Fig. 1, we illustrate the differences among instance contour adjustment (ours) and the other two related tasks, *i.e.*, image inpainting and semantic-guided (SG) image inpainting. These two exclusion rules are not enforced in image inpainting, so the generated instance contours in Fig. 1 are out of user's control, and different instances tend to be mixed to cause the ambiguous structures. The semantic-guided image inpainting methods [16,6] estimate the semantic parsing mask for the input image, infer the semantic contour for the adjusted area to complete the semantic mask, and then use the completed semantic mask to guide the inpainting process. Since the inferred semantic contour is out of user's control, these two exclusion rules are not enforced in semantic-guided image inpainting either, as demonstrated by the ambiguous structures of instances in Fig. 1. Thus, it is nontrivial to study how to effectively enforce the two exclusion rules in order to generate reasonable textures for the adjusted area (dilated or eroded) while preserving the clear and desired contours that separate different instances.

In this paper, we propose a two-stage method to leverage the structural cues to address the instance contour adjustment. The dilated area and eroded area have their respective exclusion rule, so if they exist in the same image, we will handle them separately in order to avoid the potential conflicts. It is easier to

complete the missing structural cues for the adjusted areas than completing the missing image textures, so we extract and complete two structural cues at the first stage, *i.e.*, the structure image and the depth map, and use them to guide the completion of image textures at the second stage. In order to enforce the two exclusion rules, we propose a diffusion algorithm and employ a structure reconstruction model in [14] to complete the missing structural cues for the dilated area and eroded area, respectively.

At the second stage, we propose a structure-driven CNN to generate image textures based on the structural cues completed at the first stage. By "structure-driven", we mean that both the convolution operations and the attention mechanism follow the structural cues to sample potential regions of the same instance regardless of their distribution on the image plane while passing over regions of the distracting instances. Thus, different instances will not be mixed by the CNN for the adjusted area, and the clear and desired contours will be guaranteed. The structure-driven context sampling is performed as follows: given one region on the image plane, we estimate and rank the likelihoods of its contextual regions belonging to the same instance as itself, and sample the top-ranked regions as contexts. To compute the likelihood of two regions belonging to the same instance, we consider their two affinities: *(i)* the appearance affinity based on their inclusion relationship or color distance, and *(ii)* the geometry affinity based on their depth distance.

We collect a new landscape dataset to evaluate our method, and we also establish an evaluation protocol which is beneficial to the follow-up works.

2 Related Works

Semantic-guided inpainting attracts attention recently because the semantic mask can provide the structural cues for guiding image inpainting. The AIM 2020 Challenge on Image Extreme Inpainting [12] found that the introduction of the semantic mask can both increase and decrease the performance on the inpainting task, depending on how its processing was implemented. Therefore, it is nontrivial to study how to make use of the structural cues. There are only a few methods focusing on the semantic-guided inpainting. For example, Song *et al.* [16] proposed a two-stage network, where the first stage completes the hole regions of the semantic mask, and the second stage completes the hole of the image. Liao *et al.* [6,7] proposed to estimate the semantic mask on the fly, and the estimated mask is injected back to influence the intermediate feature maps.
Coarse to fine. Many existing inpainting methods implement the coarse-to-fine pipeline with two stages. Their first stage completes the structural cues such as edge-preserved structure image [14], contour of foreground object [20], monochromic image [17], and coarse textures [22,23,24,11,13,2,26]. Their second stage directly takes as input the completed structural cues to generate image textures. However, we observe that it is hard to exert the influences of the structural cues effectively by processing them with the learnable parameters. Therefore, our structure-driven CNN uses them as the guidance to sample the

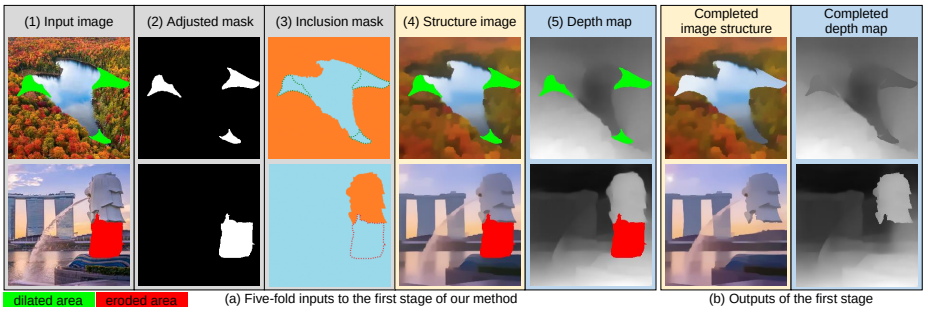


Fig. 2: **Inputs and outputs of structural cue completion (first stage).** (2) An adjusted mask is a binary mask which indicates the adjusted area. (3) An inclusion mask specifies correspondences between the adjusted area (enclosed in green/red dotted contour) and the others. Corresponding areas are set the same label, and visualized in the same color.

irregularly-distributed top-ranked regions as contexts for the convolution operation and attention mechanism, which leads to user-desired contours.

Negative influence of the void (hole) regions is a key factor downgrading the inpainting quality. In [9,24,18], various versions of partial convolution are proposed, which maintains or infers a mask to zero out values of the void regions. The definition of void regions in instance contour adjustment is more complicated because of higher demand for preserving desired contours. The void regions of a location are defined as those belonging to the distracting instances which need to be determined adaptively by structural cues.

Image extrapolation by object completion [1] is related to completing the dilated area, but several aspects differentiate this task from the instance contour adjustment. First, this task requires an input mask with the semantic label, which harms the generalization capability. Second, the inferred contour is out of user's control. Third, the object completion model cannot complete the eroded area due to the lack of clear correspondences with the external areas.

3 Structural Cues

We employ two structural cues: *(i)* structure image and *(ii)* depth map. See Fig. 2 as an example. Based on the extracted and completed structural cues, given a region on the image plane, we can estimate and rank the likelihoods of its contextual regions belonging to the same instance as itself. Thus, the convolution operation and attention mechanism can sample the top-ranked contexts precisely regardless of their distribution on the image plane. Then, we introduce the format, extraction and completion of the structural cues.

Structure image is a kind of edge-preserved smooth image which is obtained by removing the high-frequency textures while retaining the sharp edges and the low-frequency structures. Regions of an instance tend to share similar color ap-

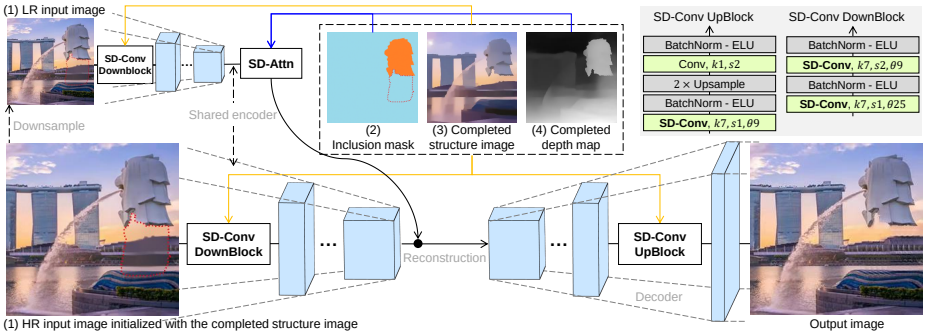


Fig. 3: **Structure-driven CNN (second stage).** The inputs are four-fold as shown by numbers in parentheses. "LR" and "HR" represent "low-resolution" and "high-resolution", respectively. The red dotted contour highlights the initialized adjusted area. SD-Conv and SD-Attn represent structure-driven convolution and attention, respectively. Orange and blue arrows indicate inputs to SD-Conv block and SD-Attn, respectively. The ellipsis represents Gated-Conv. The detailed architecture of SD-Conv blocks are attached at the top right corner. k and s denote the kernel size and stride, respectively. θ is a specialized parameter of SD-Conv denoting the number of contexts to be sampled.

pearances in a structure image. We use RTV method [21] to extract the structure image from an RGB image.

Depth map is a grayscale image in which each entry represents the relative distance of an instance surface to the camera. The depth map can be used to differentiate instances at different distances from the camera. We employ the MegaDepth method [5] to extract the depth map from a monocular RGB image.

Completion. Fig. 2 shows the inputs and outputs of the first stage of our method which completes the structural cues for the adjusted area (dilated or eroded). An inclusion mask specifies (potential) correspondences between the adjusted area and the others, and the corresponding areas are set the same label. For example, the dilated area (green) in Fig. 2 (a) corresponds to the instance area (lake), so both the dilated area and instance area are set the same label, *i.e.*, 1 (blue); the remaining area is set the label 0 (orange). The eroded area (red) in Fig. 2 (a) potentially corresponds to the area which is external to the instance area (Merlion), so both the eroded area and the external area are set the same label, *i.e.*, 1 (blue); the instance area is set the label 0 (orange). The inclusion mask is not a binary mask, which enables adjusting contours for multiple instances simultaneously.

The first stage of our method completes the structural cues for the dilated and eroded areas using different approaches because of their respective exclusion rule. Specifically, we propose a diffusion algorithm based on the iterative Gaussian blur operations which under the guidance of the inclusion mask, can propagate the structural cues from the instance area to the corresponding dilated area. Due to the lack of clear correspondences between the eroded area and the external

area, the hypothetical distribution of the external instances needs to be inferred for the eroded area. Thus, we modify the structure reconstruction model in [14] to complete the eroded area on the structure image and depth map. To avoid the interference from the instance area, we temporarily cut out the instance which will be pasted back after the completion, and complete the entire instance area as doing the eroded area. We put the details of the diffusion algorithm and the modified structure reconstruction model in the supplementary material.

4 Structure-driven CNN

Fig. 3 shows the pipeline of the structure-driven CNN. The inputs are four-fold including the two structural cues completed at the first stage. As highlighted by the red contour, the adjusted area on the input image is initialized with the corresponding area on the completed structure image which is smooth due to its nature. Thus, the structure-driven CNN is supposed to enrich the image texture details. The architecture consists of a shared encoder and a decoder with which the structure-driven convolution (SD-Conv) blocks are equipped. There is a structure-driven attention mechanism (SD-Attn) between the encoder and decoder. As in [27], in order to save the computational cost, we perform the attention estimation at low resolution.

4.1 Structure-driven convolution

Before diving into the introduction of SD-Conv, we first explain why the conventional convolutions for inpainting, *e.g.*, Gated Convolution (Gated-Conv) and Deformable Convolution (Deform-Conv), tend to mix different instances for the adjusted area, and thus cannot preserve the clear and desired contours. To this end, we prepare two baseline methods by replacing all convolutions in our method with Gated-Conv and Deform-Conv, respectively.

As illustrated in Fig. 4 (a1) and (a2), Gated-Conv and Deform-Conv sample all contexts within the receptive field regardless of whether they belong to the distracting instances. As highlighted in red, these two convolutions are prone to introduce irrelevant contexts of the distracting instances when the convolution kernels are sliding over contours separating instances. Consequently, the sampled irrelevant contexts cause the ambiguous contours between instances, as shown in Fig. 4 (a1) and (a2). The drawbacks of Gated-Conv and Deform-Conv are two-fold. First, they sample a good number of irrelevant contexts. Second, simply enlarging the kernel size might introduce more relevant contexts but it will not help to improve the ratio of relevant contexts or even worse introduce more irrelevant contexts.

In order to block the interference from the distracting instances during the context sampling process, we propose SD-Conv which can estimate and rank the likelihoods of contexts belonging to the same instance as the kernel center. As shown in Fig. 4 (a3), the SD-Conv can block the interference effectively by only sampling the top-ranked contexts.

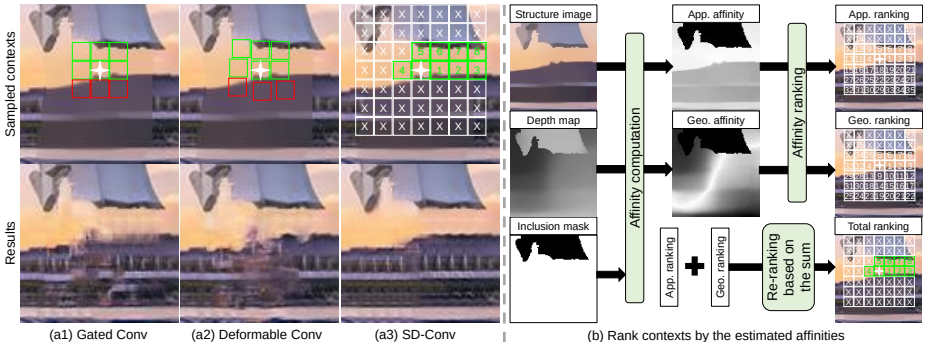


Fig. 4: **(a1-3) Sampled contexts and generation results of different convolutions.** The sampled contexts belonging to the same instance and the distracting ones are colored in green and red, respectively. The white cross marks the center of a convolution kernel. The numbers within boxes indicate the ranking of contexts and the priority of context sampling, and "X" marks the unsampled contexts or those belonging to the distracting instances. **(b) Rank the contexts by the computed affinities.** The appearance (App.) and geometry (Geo.) affinities are ranked in descending order, and these two rankings are summed for a re-ranking to obtain the total ranking of contexts. The top-ranked θ contexts are sampled. The brighter means the greater affinity. Please zoom in for details.

Affinity computation. To estimate the likelihood of a contextual region belonging to the same instance as the kernel center, as shown in Fig. 4 (b), we consider two affinities: the appearance affinity and geometry affinity.

Appearance affinity measures the appearance similarity between any contextual region and the kernel center based on the structure image, so it helps SD-Conv circumvent distracting instances of different appearances. Let i and j denote indices of the kernel center and a contextual region, and $a_{i,j}^{\mathrm{App}} \in [0,1]$ denote their appearance affinity. There are two cases for computing $a_{i,j}^{\mathrm{App}}$ depending on the location of the kernel center. If the kernel center is within the dilated area, then $a_{i,j}^{\mathrm{App}}$ can be determined directly by the clear correspondence specified by the inclusion mask $M^{\mathrm{Inc}} \in \mathbb{Z}^{H\times W}$. Otherwise, there are no clear correspondences specified in M^{Inc} for the kernel center, so we need to estimate $a_{i,j}^{\mathrm{App}}$ based on the closeness in the HSV color space [15] of the structure image which is denoted as $c_{i,j} \in [0,1]$. Let Ω denote the index set of the dilated regions. $a_{i,j}^{\mathrm{App}}$ can be computed as follows:

$$a_{i,j}^{\mathrm{App}} = \begin{cases} \delta(M_i^{\mathrm{Inc}} = M_j^{\mathrm{Inc}}), & \text{if } i \in \Omega \\ \delta(M_i^{\mathrm{Inc}} = M_j^{\mathrm{Inc}}) \cdot c_{i,j}, & \text{otherwise.} \end{cases} \tag{1}$$

$\delta(\cdot)$ is an indicator function which outputs 1 when $M_i^{\mathrm{Inc}} = M_j^{\mathrm{Inc}}$ and otherwise 0. If the kernel center is outside the dilated area where only the potential correspondences are specified, the indicator function can help exclude contextual

regions which definitely belong to the distracting instances, and $c_{i,j}$ determines $a_{i,j}^{\text{App}}$ for contextual regions with potential correspondences, which is defined as:

$$c_{i,j} = 1 - 1/\sqrt{5}\big((v_i - v_j)^2 + (s_i \cos h_i - s_j \cos h_j)^2 + (s_i \sin h_i - s_j \sin h_j)^2\big)^{\frac{1}{2}}, \quad (2)$$

where h, s and v denote values of the corresponding regions on the structure image in the HSV color space.

Geometry affinity measures the proximity of a contextual region to the kernel center. The appearance affinity cannot help circumvent distracting instances with similar colors but different textures because its computation is partially based on the structure image of which the texture details are wiped off. Yet, such distracting instances can usually be differentiated by depths in the depth map. The geometry affinity $a_{i,j}^{\text{Geo}}$ is computed as:

$$a_{i,j}^{\text{Geo}} = 1 - |M_i^{\text{Dep}} - M_j^{\text{Dep}}|/(M_{\max}^{\text{Dep}} - M_{\min}^{\text{Dep}} + \epsilon), \quad (3)$$

where M^{Dep} denotes the depth map, $M_{\max}^{\text{Dep}}$ and $M_{\min}^{\text{Dep}}$ denote the largest and smallest value in M^{Dep}. $\epsilon = 1e^{-4}$ is used to avoid dividing zero.

Affinity ranking. We perform the ranking over the computed affinities in order to determine a group of most relevant contextual regions for the kernel center. As shown in Fig. 4 (b), the affinity ranking is performed for the two affinities respectively in descending order. In order to combine the ranking for the two affinities, we sum their respective ranking, and re-rank the ranking sum in ascending order to form the total ranking which is used in Fig. 4 (a3). Finally, the top-ranked θ contexts are sampled.

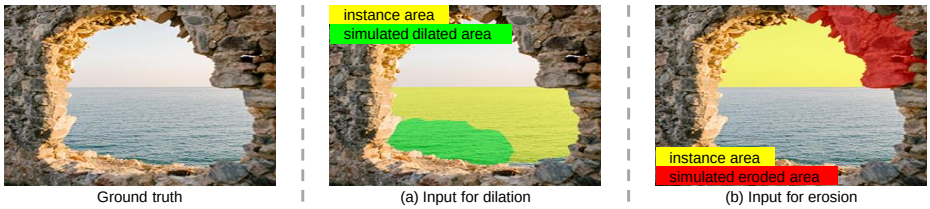
With the ranking of contexts, we can set a large kernel size for SD-Conv boldly without worrying about blending the irrelevant contexts. Yet, we do not observe clear gains for deploying the SD-Conv for the internal levels of the encoder and decoder, so we employ the conventional Gated-Conv for them which are indicated by ellipses in Fig. 3.

4.2 Structure-driven attention

The convolution operations help complete image textures by aggregating local contexts. In order to exploit the useful global contexts which are far away, Yu *et al.* proposed the contextual attention mechanism [23], which aggregates and projects the information of the contextual regions for each region according to the estimated region similarities, *i.e.*, attention weights. Let h_i and h_j denote the features at region i and j from the input feature maps, respectively. Let $s_{i,j}$ denote the attention weight of region i paid to location j, which is computed as:

$$s_{i,j} = \frac{\exp\big(\alpha \cdot \cos(h_i, h_j)\big)}{\sum_k^N \exp\big(\alpha \cdot \cos(h_i, h_k)\big)}, \quad (4)$$

where α is a hyperparameter that enlarges the range of cosine similarity $\cos(\cdot, \cdot)$, and increases the attention paid to the relevant regions. In practice, α is 10.



Ground truth (a) Input for dilation (b) Input for erosion

Fig. 5: Simulated inputs for training and evaluation.

Table 1: **Ablation study.** The higher SSIM (%) and PSNR (dB) and the lower FID, the better performance. ↑ (↓) means higher (lower) is better.

Method	Dilation			Erosion		
	FID↓	SSIM↑	PSNR↑	FID↓	SSIM ↑	PSNR↑
SD-Conv $\ominus$ a^{App} (1)	12.58	95.37	28.10	11.75	95.46	29.06
SD-Conv $\ominus$ a^{Geo} (3)	12.61	95.39	28.29	12.06	95.42	29.16
w/ Gated-Conv	12.77	95.32	28.21	11.73	95.48	29.35
w/ Deform-Conv	12.99	95.37	28.07	11.82	95.43	29.23
SD-Attn $\ominus$ a^{Geo} (3)	12.57	95.40	28.29	11.78	95.44	29.35
w/ Context-Attn	13.99	95.30	27.52	13.30	95.36	28.55
Ours	**12.43**	**95.48**	**28.32**	**11.64**	**95.53**	**29.38**

Yet, directly applying the contextual attention in our CNN leads to the ambiguity artifact as highlighted in "w/ Context-Attn" of Fig. 6 which is caused by the inaccurate attention weights. Thus, we propose SD-Attn which addresses the ambiguity artifact effectively as shown in "Ours" of Fig. 6. SD-Attn integrates the appearance affinity $a_{i,j}^{\text{App}}$ (1) and geometry affinity $a_{i,j}^{\text{Geo}}$ (3) to estimate the attention weight in order to block the interference from the distracting instances. However, we do not directly apply $a_{i,j}^{\text{App}}$, because the color closeness $c_{i,j}$ is less reliable in differentiating instances at the global level (attention) than at the local level (convolution). Therefore, we tailor $a_{i,j}^{\text{App}}$ to attention's needs:

$$a_{i,j}^{\text{App*}} = \delta(M_i^{\text{Inc}} = M_j^{\text{Inc}}) \cdot \cos(h_i, h_j), \tag{5}$$

where the appearance similarity $\cos(h_i, h_j)$ in (4) replaces $c_{i,j}$ in (2). Though $\cos(h_i, h_j)$ is observed to be more reliable than $c_{i,j}$ in measuring the appearance similarity, computing $\cos(h_i, h_j)$ requires much higher computation than $c_{i,j}$ due to the high dimensionality of h_i. Therefore, $c_{i,j}$ is more suitable for the convolution operation which is conducted much more frequently and locally than the attention mechanism. Then, the SD-Attn weight $s_{i,j}^{\text{SD}}$ is defined as:

$$s_{i,j}^{\text{SD}} = \frac{\exp\left(\alpha \cdot a_{i,j}^{\text{App*}} \cdot a_{i,j}^{\text{Geo}}\right)}{\sum_k^N \exp\left(\alpha \cdot a_{i,j}^{\text{App*}} \cdot a_{i,j}^{\text{Geo}}\right)}. \tag{6}$$

5 Experiments

Implementations. We train the structure-driven CNN using L1 loss and adversarial loss, and present its training details, detailed architecture and hyperparameters in the supplementary. We implement our method using PyTorch, and train 400 epochs with batch size 48 on 3 Nvidia P100 GPUs.
Datasets. We collect landscape images from existing datasets (COCO [8] with CC-BY 4.0 licence and ADE20K [28] with BSD license) to compose our dataset, which contains nearly 53.4K training images and 1K test images. The dataset and preparation scripts will be released upon acceptance. All images are resized to 256×256. There are two settings for evaluation, *i.e.*, "Dilation" and "Erosion". To evaluate our method quantitatively, we use the collected images as the ground truth after the instance contour adjustment, and simulate the input images by overlaying the harvested instance masks onto the image plane. In Fig. 5, we show how to simulate an input image with the dilated area and eroded area, respectively. To simulate the dilated area in Fig. 5 (a), we overlay a foreground area, and treat the instances being partially occluded as those to be dilated. To simulate the eroded area in Fig. 5 (b), we put a foreground area next to the contour of an instance, and assume that the eroded area of this instance is covered by the foreground area precisely. To increase the robustness for training, we regard different category area in the segmentation mask as the instance area to simulate the image matting process, and the overlaid foreground area is treated as the adjusted area. During testing, we only need the instance area obtained by image matting function.
Quality metrics. We adopt three widely used metrics to measure the generated image quality: Frechet Inception Distance (FID) [3], Structural Similarity Index (SSIM) [19] and Peak Signal-to-Noise Ratio (PSNR) [4]. We use FID and PSNR to measure the authenticity of the restored textures from the macroscopic and microscopic perspectives, respectively. We use SSIM to measure how well the instance contours are preserved in the generation result, so we calculate the SSIM on the image gradient level which can better reflect the similarity and clearness of contours than the RGB images do.
Efficiency metrics. We use THOP [29] to measure the parameter size (Param) and multiply-accumulate ops (MACs).

5.1 Ablation study

We disable various modules to create six baselines to study the impact of SD-Conv and SD-Attn on our method.
SD-Conv. To study the impact of appearance affinity a^{App} (1) and geometry affinity a^{Geo} (3) in determining the context ranking, we create two baselines by disabling one of them, *i.e.*, "SD-Conv $\ominus$ a^{App}" and "SD-Conv $\ominus$ a^{Geo}". Table 1 and Fig. 6 show that disabling either affinity leads to lower SSIM and ambiguities along the generated contour, which shows the necessity of these two structural cues in SD-Conv.

Table 2: **Comparison with inpainting methods.** * marks baselines taking the same five-fold inputs as our method (Fig. 2). † marks the semantic-guided inpainting baseline which removes the first stage but uses the ground-truth semantic mask to replace the semantic mask completed by its first stage; we do not show its efficiency results due to the unfair shortcut. The lower Param (M) and MACs (G) are, the better efficiency. We use green and red to highlight the efficiency for "Dilation" and "Erosion", respectively. ↑ (↓) means higher (lower) is better.

Method	Dilation			Erosion			Efficiency	
	FID↓	SSIM↑	PSNR↑	FID↓	SSIM↑	PSNR↑	Param↓	MACs↓
PEN-Net*	21.01	91.89	24.79	19.67	92.35	25.13	13.38	56.93
PEN-Net	21.61	92.10	27.05	19.55	92.61	27.62	13.37	56.81
StFlow*	19.86	91.82	26.49	18.84	92.32	26.84	92.66	271.90
StFlow	21.09	91.81	25.93	20.02	92.31	26.27	92.52	262.44
FreeForm*	13.76	92.73	28.18	12.08	93.20	29.35	16.36	117.61
FreeForm	14.48	92.53	27.78	13.09	93.00	28.63	16.17	104.34
Rethink*	13.27	92.76	27.36	12.12	93.19	27.93	130.33	138.28
Rethink	16.95	92.57	26.70	15.26	93.02	27.31	130.31	137.94
ExtInt*	20.26	92.08	26.95	18.76	92.45	27.73	16.51	5.62K
ExtInt	22.42	92.09	25.99	21.10	92.35	26.60	16.16	5.60K
DivStruct*	20.48	91.50	25.33	18.48	92.04	26.12	76.28	113.9K
DivStruct	21.09	91.47	25.02	18.92	92.00	25.79	76.26	113.1K
CRFill*	29.30	90.12	21.38	28.32	91.89	22.18	4.08	27.30
CRFill	19.19	92.60	25.91	17.25	93.06	26.61	4.05	25.25
SPG-Net*	16.64	92.34	26.31	15.17	92.82	26.91	96.17	65.16
SPG-Net†	15.68	92.39	26.97	14.07	92.90	27.67	-	-
SPG-Net	16.29	92.37	26.79	14.69	92.88	27.43	96.13	64.54
SGE-Net	14.74	92.35	27.62	12.80	92.84	28.33	73.61	212.14
Ours	**12.43**	**95.48**	**28.32**	**11.64**	**95.53**	**29.38**	0.40 47.42	8.00 270.09

We also replace the SD-Conv with Gated-Conv and Deform-Conv, respectively. Besides worse contour clearness, the generated image textures become less natural as shown in Fig. 6. This demonstrates the efficacy of SD-Conv in generating meaningful textures.

SD-Attn. The biggest difference between SD-Attn and contextual attention is the introduction of the geometry affinity a^{Geo} (3) into the attention estimation. In Fig. 6, we observe ambiguous contours after we disable a^{Geo} in SD-Attn. We also completely replace SD-Attn with the vanilla contextual attention, which leads to significant degradation in FID and PSNR in Table 1 and ambiguity artifacts in Fig. 6.

5.2 Comparison with baselines

Compared baselines include PEN-Net [25], StFlow [14], FreeForm [24], Rethink [10], ExtInt [17], DivStruct [13], CRFill [26], SPG-Net [16] and SGE-Net [6].

Fig. 6: **Ablation study.** The cyan dotted boxes highlight the artifacts.

We modify all baselines except SGE-Net to take the same five-fold inputs as ours (Fig. 2). Specifically, we modify these methods by concatenating the additional inputs with their original ones, adjusting the input dimension of their first layer, and retraining them on our dataset. The modified methods are marked with *. We cannot modify SGE-Net because it is built on a pretrained ResNet which requires exactly three input dimensions. We also show results of baselines without modification in Table 2 and Fig. 7. We prepare a stronger baseline (marked by †) for SPG-Net by replacing the semantic mask completed by its first stage with the ground-truth semantic mask which contains the user-desired semantic distribution for the adjusted area. We do not prepare such a stronger baseline for SGE-Net because its inferred semantic mask needs to be filled with activations ranging from the negative infinity to the positive infinity.

Analysis. Table 2 shows that after introducing the additional inputs, most baselines achieve better or comparable performance, but our method still outperforms them. The SSIM of all baselines are smaller than 93.3%, while our method achieves nearly 95.5% in both settings, which demonstrates the effectiveness of our method in preserving clear and user-desired contours. Such an advantage is also reflected in Fig. 7 where baselines yield non-negligible artifacts when dilating or eroding regions.

Only the performance of CRFill [26] drops drastically after the modification. As a two-stage method, its first stage is a simple CNN for generating coarse textures, and the second stage refines coarse textures. Since the additional inputs are given at the first stage, we argue that its first stage is too simple to learn processing the inputs containing such rich structural cues, and thus it causes inferior coarse textures which are too hard to be recovered by its second stage.

Efficiency & **limitation.** Table 2 shows that our method is efficient for "Dilation", but is cumbersome for "Erosion". This is because we use different ap-

Table 3: User study results. Our method outperforms other approaches with the highest scores on both dilation and erosion experiments.

Experiment	FreeFrom* [24]	Rethink* [10]	SPG-Net† [16]	SGE-Net [16]	Ours
Dilation	10.66%	14.41%	22.50%	20.41%	**32.01**%
Erosion	9.63%	16.47%	23.28%	21.37%	**29.25**%

proaches to complete the structural cues for these two settings in the first stage (see § 3 for the motivation). For "Dilation", we propose a diffusion algorithm based on the efficient Gaussian blur (see the supplementary for details). For "Erosion", we modify the structure reconstruction model in [14] which is heavy. Our ultimate goal is to deploy our method on phones, so we need to optimize the efficiency for "Erosion". The MACs of ExtInt and DivStruct are significantly higher than the others, because they adopt iterative processing strategies.

5.3 User Study

We select four baselines with leading performance in the quantitative evaluation as candidates for our user study. We conduct two experiments to evaluate whether our results are more favored by human observers than other methods. Participants are shown an original image with instance with dilation/erosion area, in addition to five generated results, and asked to choose the result that matches best with the dilation/erosion area. In each experiment, 100 images are randomly selected from the testing set. Experiments are published on Amazon Mechanical Turk (AMT) and each experiment is completed by 25 participants. As shown in Table. 3, our method achieves highest scores in both experiments.

6 Conclusion

We study instance contour adjustment. The first stage of our method extracts and completes the structural cues within the adjusted contours. We further propose a structure-driven CNN for the second stage which completes the image textures based on the completed structural cues. There are mainly two novel modules, SD-Conv and SD-Attn, of which the redesigned sampling strategy can estimate and rank the relevance of the contexts based on the structural cues, and sample the top-ranked contexts regardless of their distribution on the image plane. Our method could generate meaningful textures while preserving clear and user-desired contours.

Acknowledgements

This project is supported by National Natural Science Foundation of China under Grant No. 62136001. We thank Wenbo Li for the advise and discussion for this project.

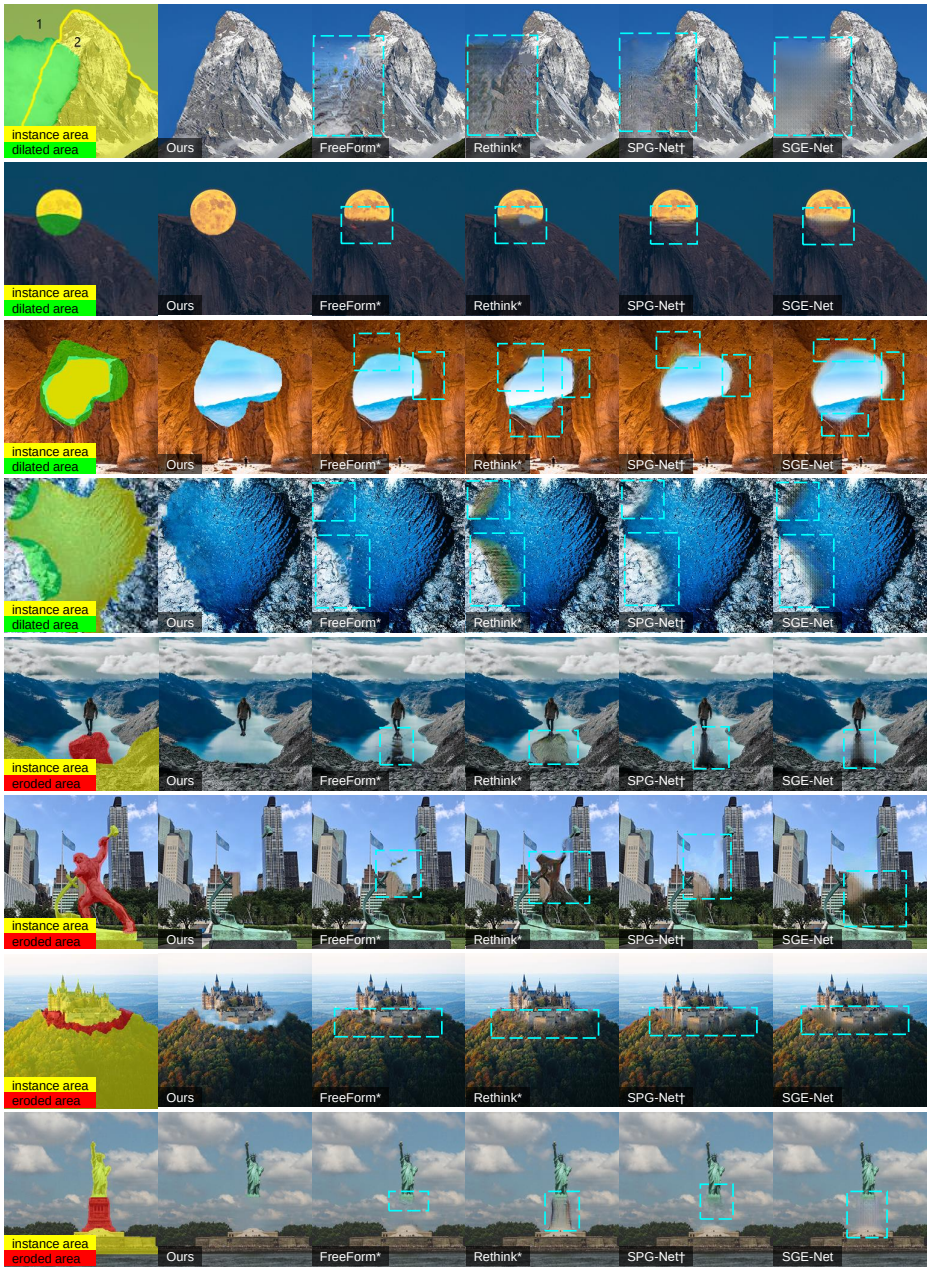


Fig. 7: **Qualitative comparisons.** The cyan dotted boxes highlight the artifacts. Due to the space limit, we only show comparison with four methods that with leading performance in quantitative evaluation.

References

1. Bowen, R.S., Chang, H., Herrmann, C., Teterwak, P., Liu, C., Zabih, R.: Oconet: Image extrapolation by object completion. In: CVPR (2021) 4
2. Guo, X., Yang, H., Huang, D.: Image inpainting via conditional texture and structure dual generation. In: ICCV (2021) 3
3. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NIPS (2017) 10
4. Huynh-Thu, Q., Ghanbari, M.: Scope of validity of psnr in image/video quality assessment. Electronics letters (2008) 10
5. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: CVPR (2018) 5
6. Liao, L., Xiao, J., Wang, Z., Lin, C., Satoh, S.: Guidance and evaluation: Semantic-aware image inpainting for mixed scenes. In: ECCV (2020) 2, 3, 11
7. Liao, L., Xiao, J., Wang, Z., Lin, C., Satoh, S.: Image inpainting guided by coherence priors of semantics and textures. In: CVPR (2021) 3
8. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: ECCV (2014) 10
9. Liu, G., Reda, F.A., Shih, K.J., Wang, T., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: ECCV (2018) 4
10. Liu, H., Jiang, B., Song, Y., Huang, W., Yang, C.: Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In: ECCV (2020) 11, 13
11. Liu, H., Wan, Z., Huang, W., Song, Y., Han, X., Liao, J.: PD-GAN: probabilistic diverse GAN for image inpainting. In: CVPR (2021) 3
12. Ntavelis, E., Romero, A., Bigdeli, S., Timofte, R., Hui, Z., Wang, X., Gao, X., Shin, C., Kim, T., Son, H., Lee, S., Li, C., Li, F., He, D., Wen, S., Ding, E., Bai, M., Li, S., Zeng, Y., Lin, Z., Yang, J., Zhang, J., Shechtman, E., Lu, H., Zeng, W., Ni, H., Cai, Y., Li, C., Xu, D., Wu, H., Han, Y., Uddin, S.M.N., Jang, H.W., Ahmed, S.H., Yoon, J., Jung, Y.J., Li, C., Liu, Z., Wang, L., Siu, W., Lun, D.P., Suin, M., Purohit, K., Rajagopalan, A.N., Narang, P., Mandal, M., Chauhan, P.S.: AIM 2020 challenge on image extreme inpainting. In: ECCVW (2020) 3
13. Peng, J., Liu, D., Xu, S., Li, H.: Generating diverse structure for image inpainting with hierarchical VQ-VAE. In: CVPR (2021) 3, 11
14. Ren, Y., Yu, X., Zhang, R., Li, T.H., Liu, S., Li, G.: Structureflow: Image inpainting via structure-aware appearance flow. In: ICCV (2019) 3, 6, 11, 13
15. Smith, J.R., Chang, S.: Visualseek: A fully automated content-based image query system. In: ACM MM (1996) 7
16. Song, Y., Yang, C., Shen, Y., Wang, P., Huang, Q., Kuo, C.J.: Spg-net: Segmentation prediction and guidance network for image inpainting. In: BMVC (2018) 2, 3, 11, 13
17. Wang, T., Ouyang, H., Chen, Q.: Image inpainting with external-internal learning and monochromic bottleneck. In: CVPR (2021) 3, 11
18. Wang, W., Zhang, J., Niu, L., Ling, H., Yang, X., Zhang, L.: Parallel multi-resolution fusion network for image inpainting. In: ICCV (2021) 4
19. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. TIP (2004) 10
20. Xiong, W., Yu, J., Lin, Z., Yang, J., Lu, X., Barnes, C., Luo, J.: Foreground-aware image inpainting. In: CVPR (2019) 3

21. Xu, L., Yan, Q., Xia, Y., Jia, J.: Structure extraction from texture via relative total variation. TOG (2012) 5
22. Yi, Z., Tang, Q., Azizi, S., Jang, D., Xu, Z.: Contextual residual aggregation for ultra high-resolution image inpainting. In: CVPR (2020) 3
23. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: CVPR (2018) 3, 8
24. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: ICCV (2019) 3, 4, 11, 13
25. Zeng, Y., Fu, J., Chao, H., Guo, B.: Learning pyramid-context encoder network for high-quality image inpainting. In: CVPR (2019) 11
26. Zeng, Y., Lin, Z., Lu, H., Patel, V.M.: Cr-fill: Generative image inpainting with auxiliary contextual reconstruction. In: ICCV (2021) 2, 3, 11, 12
27. Zeng, Y., Lin, Z., Yang, J., Zhang, J., Shechtman, E., Lu, H.: High-resolution image inpainting with iterative confidence feedback and guided upsampling. In: ECCV (2020) 6
28. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ADE20K dataset. In: CVPR (2017) 10
29. Zhu, L.: Thop: Pytorch-opcounter. https://github.com/Lyken17/pytorch-OpCounter 10