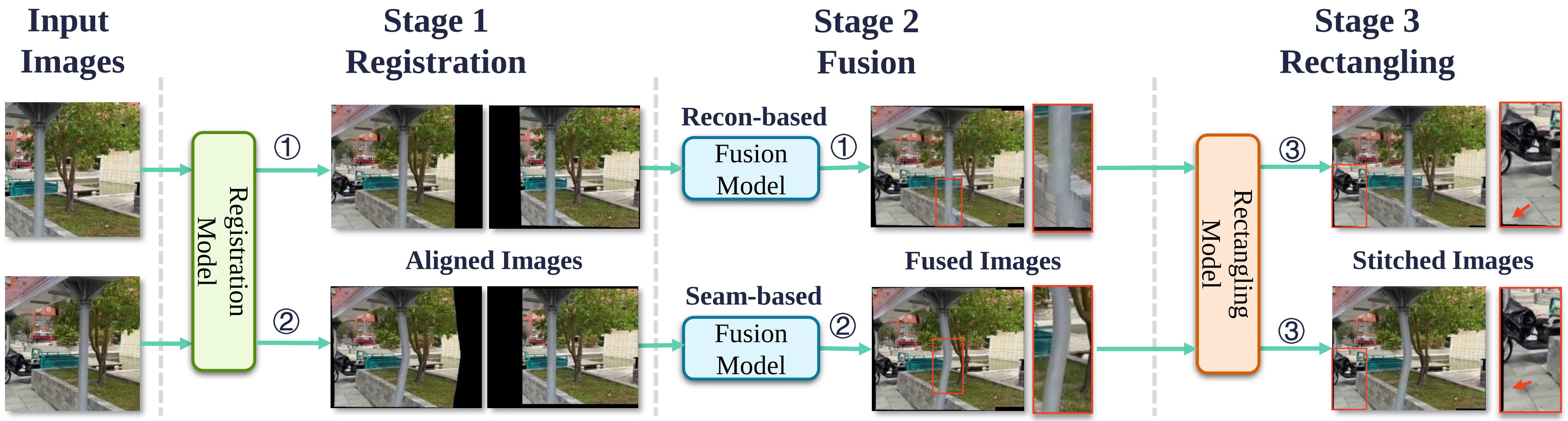
Reconstructing the Image Stitching Pipeline: Integrating Fusion and Rectangling into a Unified Inpainting Model

Ziqi Xie, Weidong Zhao, Xianhui Liu, Jian Zhao, Ning Jia

Code: https://github.com/yayoyo66/SRStitcher

Existing image stitching pipeline architecture and problems

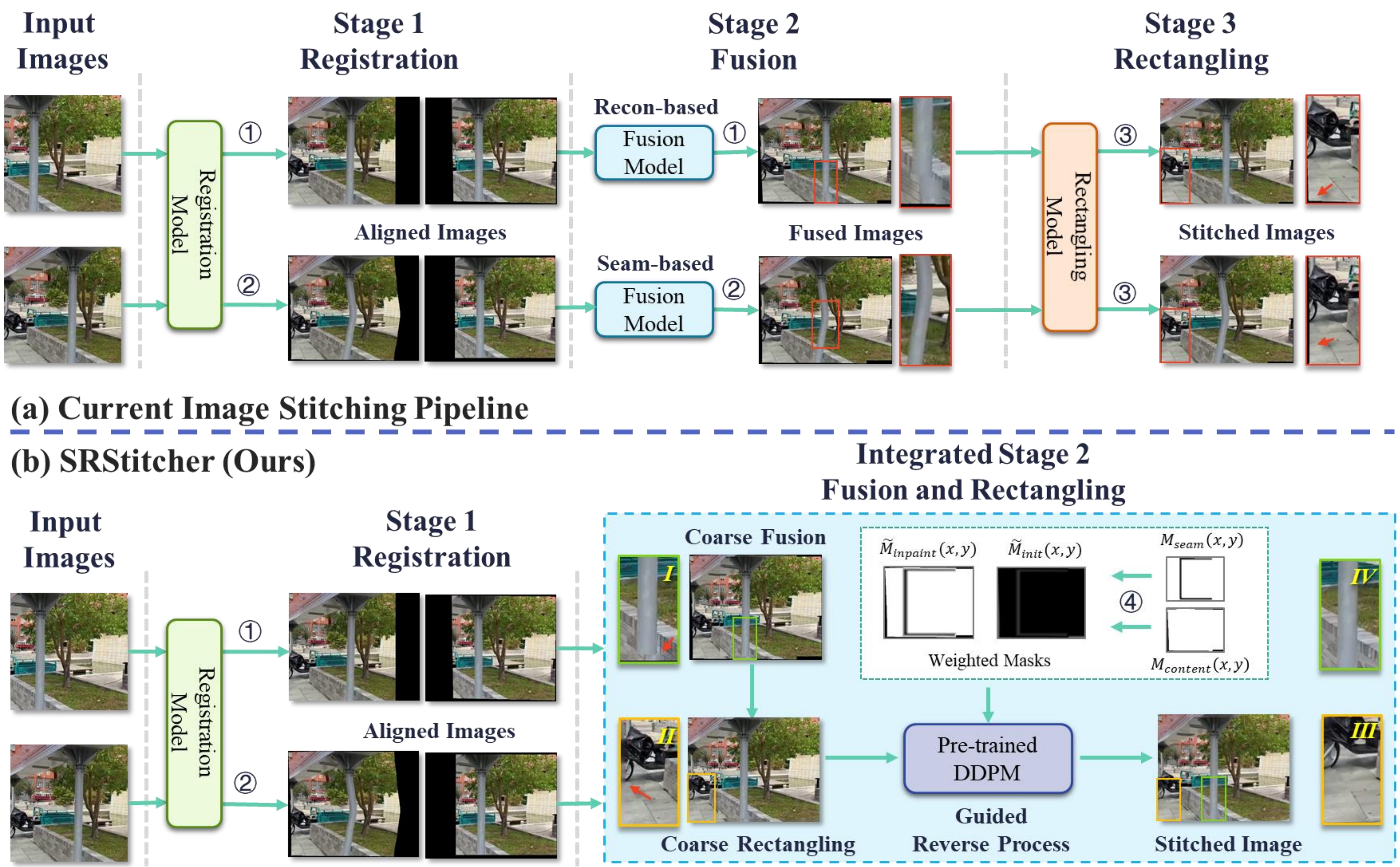
1. **Parameters Optimization**: Each stage requires training the model separately, which increases the complexity of training and optimization of the overall model.
2. **Cascading Errors**: Errors generated in the previous stage propagate to subsequent stages, and existing methods lack robustness to such error propagation. Especially, the existing rectangling methods are all based on the assumption that the images generated in the previous stage do not have any errors.



Motivation

Fusion model: Recon-based fusion methods suffer from artifacts when registration errors occur, and seam-based fusion methods cause significant distortions. ***Is there a third idea, such as applying a powerful generative model to modify the wrong fused image regions, i.e. Inpainting the wrong image.***

Rectangling model: If the fusion task can be expressed by a inpainting model, and the rectangling task is essentially a inpainting task: ***can the two tasks be unified into a inpainting model?***



TONGJI UNIVERSITY

Registration parameterization.

Suppose two input images: $I_l(x,y), I_r(x,y) \in \mathbb{R}^{H\times W}$

$\mathcal{H}$ is a 3x3 homography matrix, the stitched width and height can be obtained by:

$$W^* = \max_{k\in(1,2,3,4)} \{x_k^w, x_k^l\} - \min_{k\in(1,2,3,4)} \{x_k^w, x_k^l\},$$

$$H^* = \max_{k\in(1,2,3,4)} \{y_k^w, y_k^l\} - \min_{k\in(1,2,3,4)} \{y_k^w, y_k^l\}, \qquad \text{where, } (x_k^w, y_k^w) = \mathcal{H}\times[x_k^r, y_k^r, 1]^T.$$

Use warping function $\varphi(\cdot)$ to get the aligned images by input images:

$$I_{wl}(x,y), I_{wr}(x,y) = \varphi(I_l(x,y), \mathtt{I}), \varphi(I_r(x,y), \mathcal{H}),$$

Replace the input images by all-one matrixes，get the masks: $M_{wl}(x,y), M_{wr}(x,y)$

Parameterization: Inpainting-based fusion model

The coarse fusion image is obtained by superimposing the less distorted image on the more distorted image:

$$I_{CF}(x,y) = I_{wl}(x,y) + I_{wr}(x,y) \odot (1 - (M_{wl}(x,y) \& M_{wr}(x,y))),$$

Coarse Fusion

Use seam mask M_{seam} to define the inpainting regions:

$$\begin{aligned} M_{seam}(x,y) = {} & \texttt{Dilation}(M_{wl}(x,y), K_s) \oplus M_{wl}(x,y) \vee \\ & \texttt{Erosion}((M_{wl}(x,y), K_s) \oplus M_{wl}(x,y) \& M_{wr}(x,y), \end{aligned}$$

The parameterization of inpainting-based fusion model can be defined as：

$$\hat{I}_{CF}(x,y) = I_{CF}(x,y) \odot (1 - M_{seam}(x,y)) + f_\theta(I_{CF}(x,y)) \odot M_{seam}(x,y).$$

Parameterization: Inpainting-based rectangling model

TONGJI UNIVERSITY

The parameterization of inpainting-based rectangling model can be defined as：

$$\hat{I}_{CR}(x,y) = I_{CF}(x,y) \odot (1 - M_{content}(x,y)) + f_\theta(I_{CF}(x,y)) \odot M_{content}(x,y),$$

$$\text{where, } M_{content}(x,y) = M_{wl}(x,y) \vee M_{wr}(x,y).$$

The inpainting-based unified model can be defined as：

$$\hat{I}_{CFR}(x,y) = I_{CF}(x,y) \odot (1 - M_{inpaint}(x,y)) + f_\theta(I_{CF}(x,y)) \odot M_{inpaint}(x,y),$$

$$\text{where, } M_{inpaint}(x,y) = M_{seam}(x,y) \vee M_{content}(x,y).$$

This model only defines the image region that needs to be repainted. What we want is that: **the inpainting in the fusion region should consider the existing content of the coarse fusion image, and the inpainting in the rectangling region has higher intensity but does not deviate from the surrounding image semantics.**

The reverse process of the diffusion model

Background: DDPM reverse process

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)), t \in (1, T),$$

$\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ are the parameters of the Gaussian Markov chain at step t

DDPM-based unified inpainting model

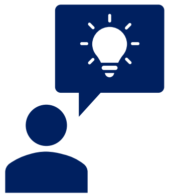
$$\hat{\mathbf{x}}_{t-1} = \mathbf{x}_0 \odot (1 - M_{inpaint}(x, y)) + \mathbf{x}_{t-1} \odot M_{inpaint}(x, y),$$

where, $\mathbf{x}_0 = \mathcal{E}(I_{CF}(x, y))$, and $\mathbf{x}_{t-1} \sim \mathcal{N}(\mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$.

Why use diffusion model

1. Modifying incorrectly fused image content requires a generative model with extremely strong generalization capabilities.

2. The reverse process of the diffusion model is a gradual modification process, which can achieve exactly the different intensity modifications we need for different regions.



Weighted initial mask $\widetilde{M}_{init}(x, y)$

$$\widetilde{M}_{init}(x,y) = \frac{\mathtt{DT}(M_{seam}(x,y), K_g) \times \epsilon_1}{\max \mathtt{DT}(M_{seam}(x,y), K_g)} \oplus \frac{\mathtt{DT}(M_{content}(x,y), K_g) \times \epsilon_2}{\max \mathtt{DT}(M_{content}(x,y), K_g)},$$

where, $\mathtt{DT}(\cdot)$ is the distance transform operation [36] with kernel size K_g, ϵ_1 and ϵ_2 are hyper-parameters.

Weighted inpainting mask $\widetilde{M}_{inpaint}(x, y)$

$$\widetilde{M}_{inpaint}(x,y) = M_{content} \vee (1 - \mathtt{DT}(M_{seam}(x,y), K_g)).$$

The role of two weight masks

1. **Weighted initial mask**: how much of the initial information of the input image to keep.

2. **Weighted inpainting mask**: In the reverse process, it is mapped into multiple sub-masks, and different sub-masks are used to control the inpainting intensity at different step t.

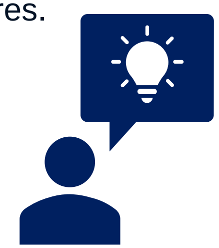
Weight mask guided reverse process

Algorithm 1 **W**eighted **M**ask **G**uided **R**everse **P**rocess (**WMGRP**)

Input: Coarse Fusion image $I_{CF}(x, y)$; Inference steps N; Radius R;
Weighted initial mask $\widetilde{M}_{init}(x, y)$; Weighted inpainting mask $\widetilde{M}_{inpaint}(x, y)$
prompt $p \leftarrow$ "" ▷ Our method does not require prompt guidance
$I_{CFR}(x, y) \leftarrow \texttt{Telea}(I_{CF}(x, y), M_{content}(x, y), R)$ ▷ Coarse rectangling
$\mathbf{x}_N \leftarrow \mathcal{E}(I_{CFR}(x, y))$ ▷ Encode image
// Based on the inpainting model, so there is a little difference here with the Eq. 9
$\mathbf{x}_0 \leftarrow \mathcal{E}(I_{CFR}(x, y) \odot \widetilde{M}_{init}(x, y))$
$\widetilde{M}^{small}_{inpaint}(x, y), \widetilde{M}^{small}_{init}(x, y) \leftarrow \texttt{DownSample}(\widetilde{M}_{inpaint}(x, y), \widetilde{M}_{init}(x, y))$
$\mathbf{x}'_N \leftarrow \texttt{AddNoise}(\mathbf{x}_N, N)$
$\hat{\mathbf{x}}_N \leftarrow \texttt{Concat}(\mathbf{x}'_N, \widetilde{M}^{small}_{init}(x, y), \mathbf{x}_0)$
for $t = N - 1, \cdots, 0$ **do** ▷ Reverse process
  $\mathbf{x}'_t \leftarrow \texttt{DeNoise}(\hat{\mathbf{x}}_{t+1}, p, t)$
  $\widetilde{M}^{small}_t(x, y) \leftarrow 1 - (\widetilde{M}^{small}_{inpaint}(x, y) \preceq \frac{N-t}{N})$ ▷ $\preceq$ means element-wise less-than
  $\hat{\mathbf{x}}_t \leftarrow \texttt{Concat}(\mathbf{x}'_t, \widetilde{M}^{small}_t(x, y), \mathbf{x}_0)$
end for
$\hat{I}_{CFR}(x, y) \leftarrow \texttt{ImageDecoder}(\hat{\mathbf{x}}_0)$ ▷ Decode image
Output: $\hat{I}_{CFR}(x, y)$

WMGRP pseudo-code

This pseudo-code is based on a pre-trained Stable diffusion inpainting model, our method does not require any training and fine-tuning, and is general over existing mainstream diffusion model architectures.



Experiment settings

TONGJI UNIVERSITY

Dataset UDIS-D

Baselines

Table 1: Statistics of related works and details of comparison baselines.

(a) Statistics of related works.

Work	Stage1	Stage2	Stage3
VFISNet [30]	✓	✓	✗
EPISNet [34]	✓	✓	✗
UDIS [31]	✓	✓	✗
UDIS++ [33]	✓	✓	✗
Dseam [11]	✗	✓	✗
Jiang et al. [22]	✓	✓	✗
LBHomo [21]	✓	✗	✗
RHWF [7]	✓	✗	✗
HomoGAN [18]	✓	✗	✗
DR [32]	✗	✗	✓

(b) Details of comparison baselines.

Baseline	Stage1 and 2	Stage3
UDIS+DR	UDIS	DR
UDISplus+DR	UDIS++	DR
UDIS+Lama	UDIS	Lama
UDISplus+Lama	UDIS++	Lama
UDIS+SD1.5	UDIS	SD1.5
UDISplus+SD1.5	UDIS++	SD1.5
UDIS+SD2	UDIS	SD2
UDISplus+SD2	UDIS++	SD2

Metrics Measure the quality of the stitched image (NR-IQA): HIQA (2020CVPR) CLIPIQA(2023AAAI)

Measure the Content Consistency Score, CCS:

$$CCS = (CCS_n + CCS_g)/2$$

$$CCS_n = \texttt{cosine}(\Upsilon(\texttt{Split}(I_{Stitched}(x,y),n)), \Upsilon(\texttt{Split}(I_{Fusion}(x,y),n)))$$

$$CCS_g = \texttt{cosine}(\Upsilon(I_{Stitched}(x,y), \Upsilon(I_l(x,y), I_r(x,y)). \qquad \Upsilon(\cdot) = \texttt{Bert}(\texttt{CoCa}(\cdot))$$

Quantitative evaluation

Table 2: Quantitative results. The best and second-best results are highlighted by **red** and blue. ⋆ refers to the inference results of this method are not affected by seed. † means the inference results of this method are affected by the seed. We tested the results five times by varying the seed, taking the average and standard deviation.

Method	$UDIS-D_{test}$			$UDIS-D_{train}$		
	HIQA ↑	CLIPIQA ↑	CCS(%)↑	HIQA ↑	CLIPIQA ↑	CCS(%) ↑
UDIS+DR⋆	42.53	28.33	89.35	45.31	31.29	90.02
UDISplus+DR⋆	45.98	31.24	88.45	49.87	33.47	90.69
UDIS+Lama⋆	42.55	27.17	84.99	45.63	30.15	86.70
UDISplus+Lama⋆	46.57	31.48	87.73	51.28	33.29	86.12
UDIS+SD1.5†	42.60 ± 2.24	28.03 ± 2.84	87.42 ± 1.08	48.59 ± 1.18	28.57 ± 0.89	87.74 ± 1.36
UDISplus+SD1.5†	46.45 ± 1.11	27.13 ± 1.85	87.16 ± 1.61	50.89 ± 2.20	30.16 ± 1.46	88.12 ± 1.35
UDIS+SD2†	42.84 ± 1.05	28.00 ± 0.89	85.97 ± 1.33	47.15 ± 1.33	34.31 ± 0.95	85.72 ± 1.55
UDISplus+SD2†	46.98 ± 1.43	31.23 ± 2.18	89.37 ± 1.23	51.49 ± 1.74	34.26 ± 1.24	91.18 ± 1.35
SRStitcher Variants						
SRStitcher-S†	45.66 ± 0.89	32.08 ± 0.91	85.91 ± 0.74	51.73 ± 0.56	35.23 ± 0.79	87.32 ± 0.81
SRStitcher-U†	43.89 ± 1.01	28.35 ± 0.66	85.81 ± 1.01	48.18 ± 0.55	31.38 ± 0.74	86.33 ± 0.53
SRStitcher-C†	46.57 ± 0.89	31.34 ± 0.76	89.47 ± 0.71	52.73 ± 0.74	34.53 ± 0.85	91.41 ± 0.84
SRStitcher†	**47.82 ± 0.55**	**33.25 ± 0.57**	**91.15 ± 0.52**	**54.74 ± 0.63**	**37.52 ± 0.68**	**93.29 ± 0.45**

SRStitcher-S

based on

Stable-Diffusion-2 model

SRStitcher-U

based on

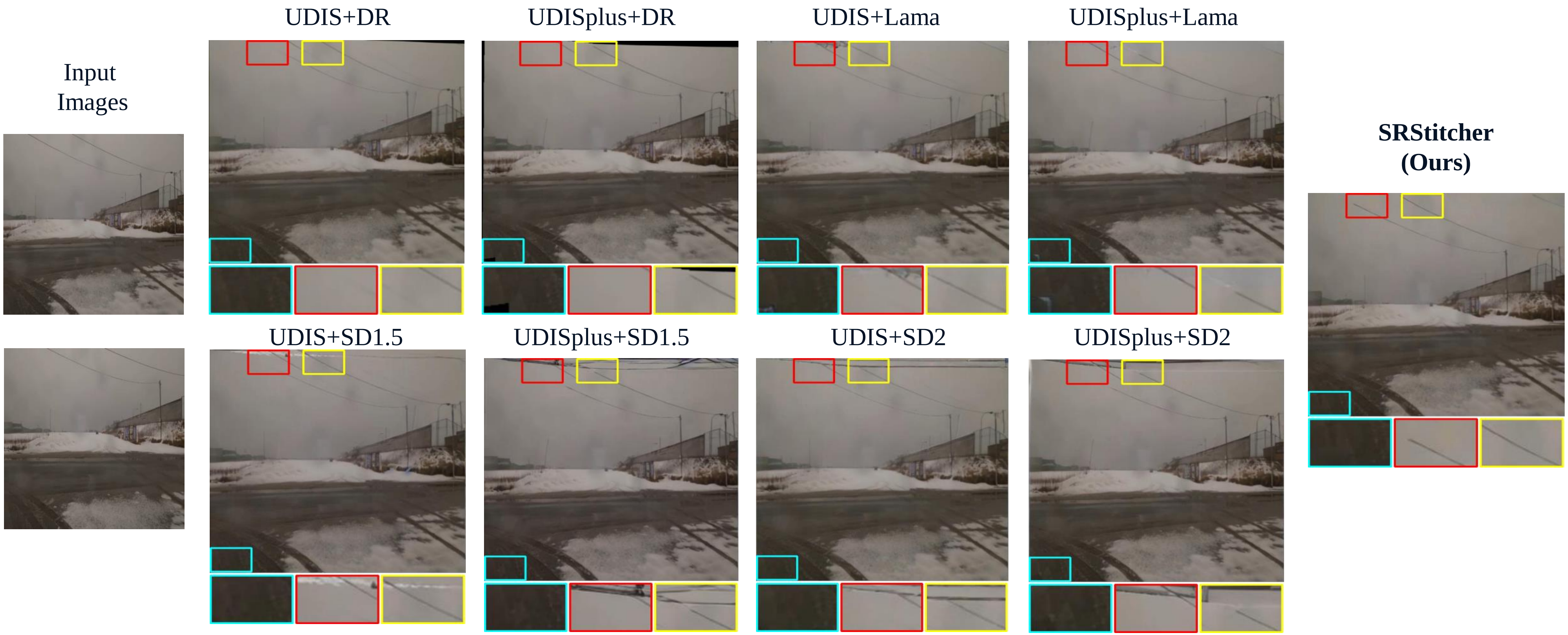
Stable-Diffusion-2-1-Unclip model

SRStitcher-C

based on

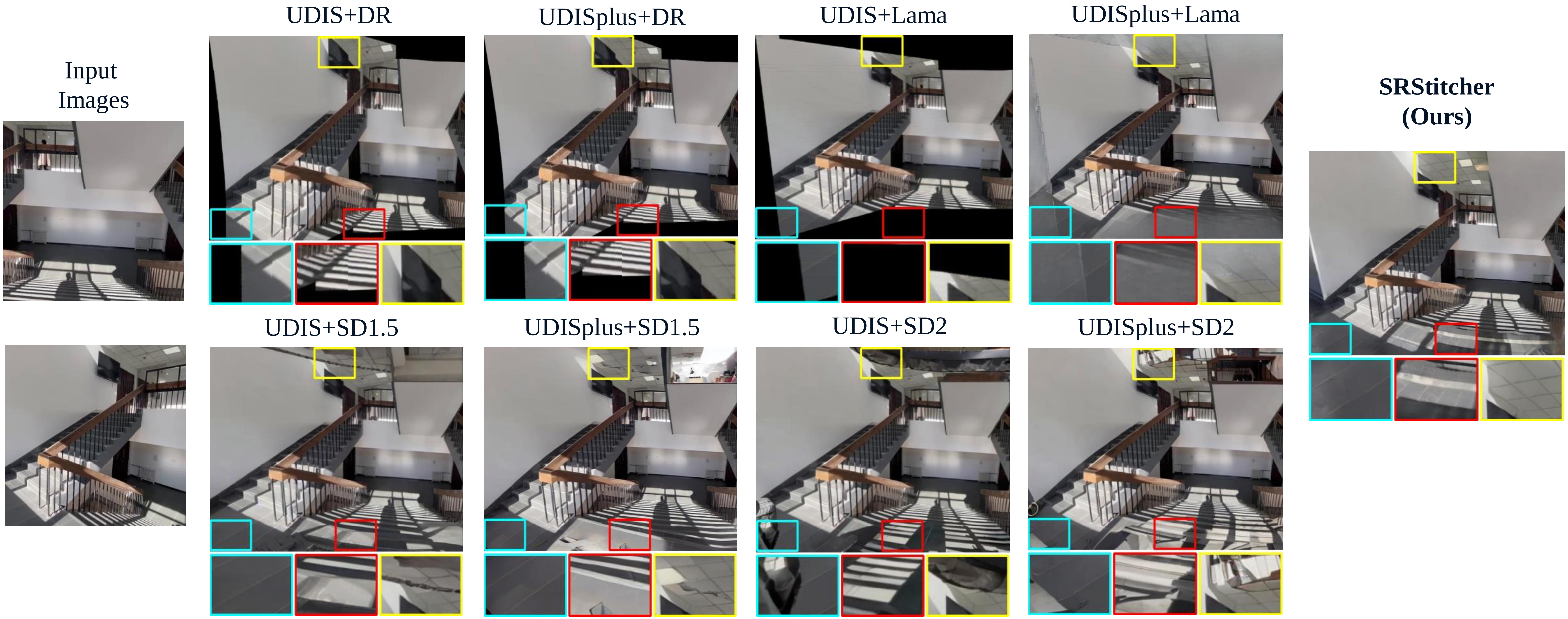
Stable-Diffusion-control-v11p-sd15-inpaint model

Qualitative evaluation

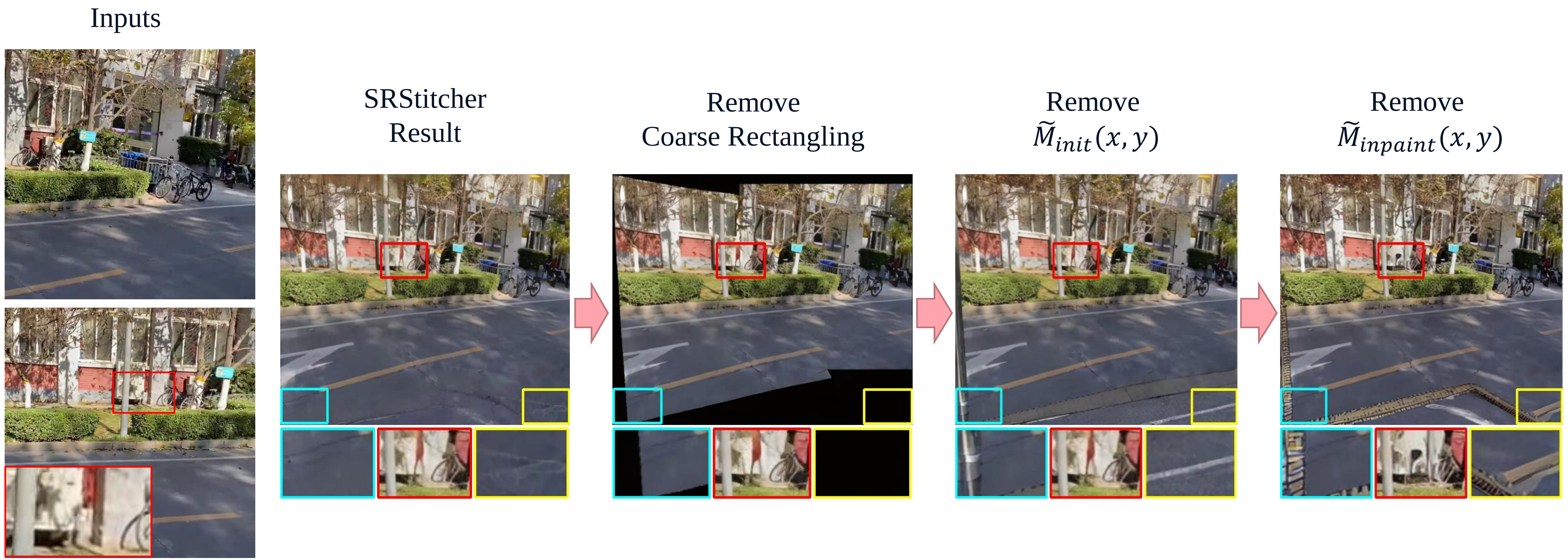


Qualitative evaluation

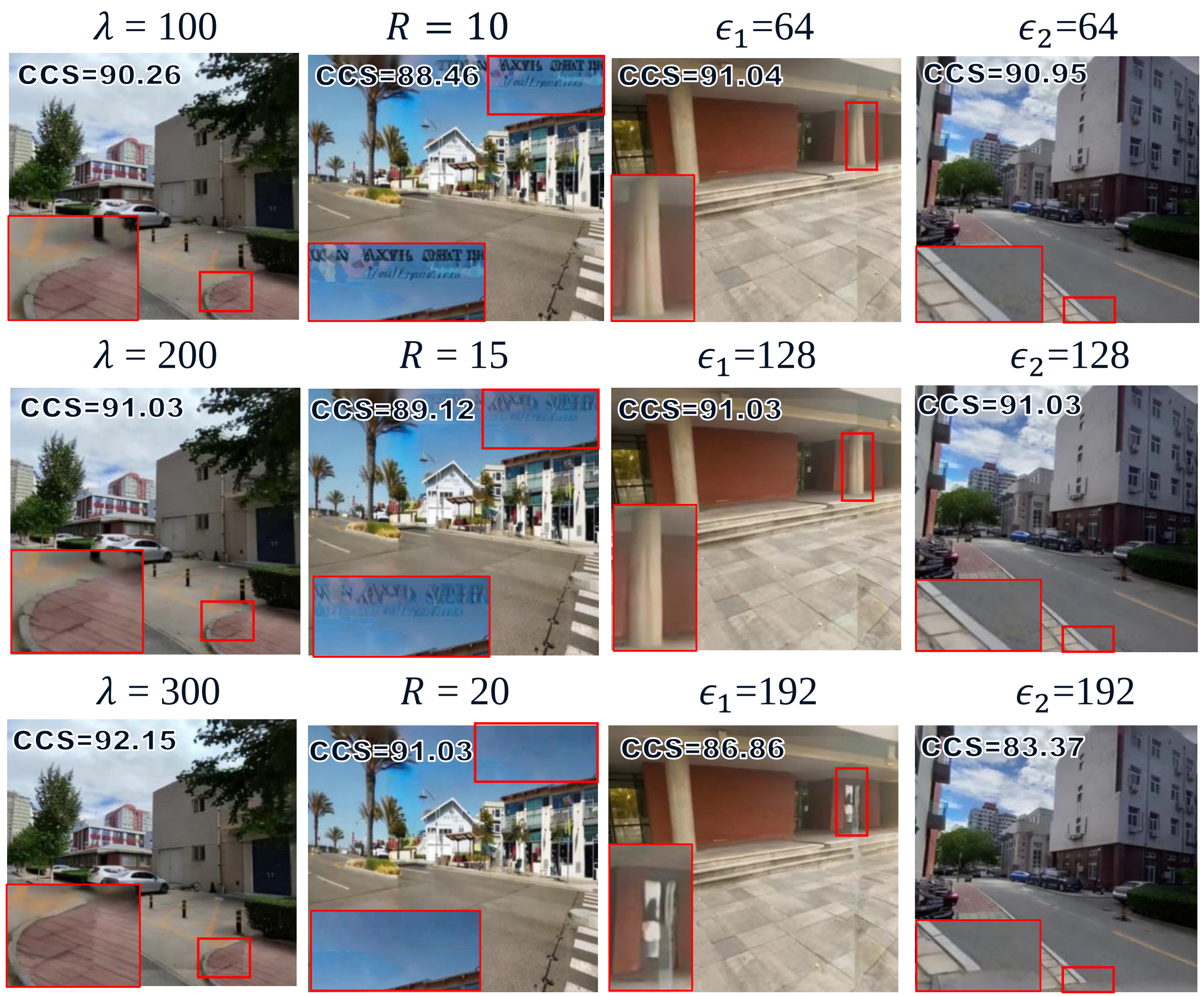
Qualitative evaluation



Ablation of modules



Ablation of hyper-parameters



$$M_{seam} = \texttt{Dilation}(M_{wl}(x,y), K_s) \oplus M_{wl}(x,y) \vee \texttt{Erosion}((M_{wl}(x,y), \boxed{K_s}) \oplus M_{wl}(x,y) \& M_{wr}(x,y),$$

$$\boxed{K_s} \rightarrow \lceil W^*/\lambda \rceil \times \delta,$$

4: $I_{CFR}(x,y) \leftarrow \texttt{Telea}(I_{CF}(x,y), M_{content}(x,y), \boxed{R})$

$$\widetilde{M}_{init}(x,y) = \frac{\texttt{DT}(M_{seam}(x,y), K_g) \times \boxed{\epsilon_1}}{\max \texttt{DT}(M_{seam}(x,y), K_g)} \oplus \frac{\texttt{DT}(M_{content}(x,y), K_g) \times \boxed{\epsilon_2}}{\max \texttt{DT}(M_{content}(x,y), K_g)},$$

Discussion and conclusion

This paper proposes:

- **Simple and Robust Stitcher (SRStitcher)**：a more streamlined and robust image stitching pipeline.
- **Redefinition:** We redefine the problems of fusion and rectangling in the image stitching pipeline and unify them into an image inpainting model that is more robust to registration errors**.**
- **Weighted mask-guided reverse process:** We design a weighted mask-guided reverse process to precisely control the inpainting strength of different regions during the generation of large-scale diffusion models, enabling inference to solve two tasks at once without additional supervision data.

Open issues:

- Obvious seams may appear when the color difference of the stitched image is large. **Solution:** dynamic parameter setting
- Local blur problem. **Solution:** Fine-tuning the model
- **Integrated the registration stage into the unified model. Solution:** Diffusion Features (DIFT)

Thanks!