

TwinTex: Geometry-aware Texture Generation for Abstracted 3D Architectural Models (Supplementary Material)

WEIDAN XIONG, Shenzhen University, China

HONGQIAN ZHANG, Shenzhen University, China

BOTAO PENG, Shenzhen University, China

ZIYU HU, Guangdong Artificial Intelligence and Digital Economy Laboratory (SZ), Shenzhen University, China

YONGLI WU, Guangdong Artificial Intelligence and Digital Economy Laboratory (SZ), Shenzhen University, China

JIANWEI GUO, MAIS, Institute of Automation, Chinese Academy of Sciences, China

HUI HUANG*, Shenzhen University, China

ACM Reference Format:

Weidan Xiong, Hongqian Zhang, Botao Peng, Ziyu Hu, Yongli Wu, Jianwei Guo, and Hui Huang. 2023. TwinTex: Geometry-aware Texture Generation for Abstracted 3D Architectural Models (Supplementary Material).

ACM Trans. Graph. 42, 6 (December 2023), 14 pages. <https://doi.org/10.1145/3618328>

1 TWINTEX PLUG-IN OF HOUDINI

We have implemented our TwinTex as a customized plug-in of Houdini. Users can use our tool by adding a TwinTex node. After that, users can load all the required inputs by specifying their location and adjust parameters in the user interface. Fig. 1 shows the texturing results of Highrise in Houdini with our TwinTex tool by simply click the *Start* button. Users can also choose to run a single step at a time by clicking the relative *Apply* button. We also provide progress bars to visualize the rate of texturing progress and visualize the intermediate result of the texture map in the console. The texturing result will be applied to the model and visualized automatically in the viewing panel once the texture mapping is finished. We provide the function for users to save the texture maps for possible later uses in other applications. Readers can refer to our video for more information on our TwinTex tool.

2 STATISTICS ON IMAGE COLLECTION

For all examples shown in this paper, we visualized the statistics of the aerial drones collecting our input images in Fig. 2. The position of a camera is visualized as a sphere, while the viewing direction is

*Corresponding author: Hui Huang (hhzhixian@gmail.com)

Authors' addresses: Weidan Xiong, xiongweidan@gmail.com, Shenzhen University, China; Hongqian Zhang, Shenzhen University, China; Botao Peng, Shenzhen University, China; Ziyu Hu, Guangdong Artificial Intelligence and Digital Economy Laboratory (SZ), Shenzhen University, China; Yongli Wu, Guangdong Artificial Intelligence and Digital Economy Laboratory (SZ), Shenzhen University, China; Jianwei Guo, MAIS, Institute of Automation, Chinese Academy of Sciences, China; Hui Huang, hhzhixian@gmail.com, Shenzhen University, College of Computer Science & Software Engineering, China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM. 0730-0301/2023/12-ART \$15.00 <https://doi.org/10.1145/3618328>

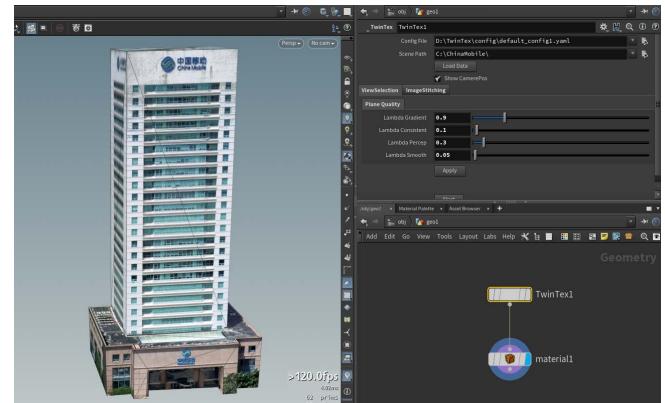


Fig. 1. Our TwinTex node in Houdini. Users can specify the location of inputs/outputs and parameters in the user interface.

visualized as a line pointing from its location. The paths of all the aerial drones are specially designed for better reconstructing dense models [Zhang et al. 2021; Zhou et al. 2020], not for the purpose of high-quality texturing. A notable example is the School whose images are shot sparsely and far away from the building. The variety of drone paths also demonstrates the robustness and generalization ability of our approach.

The statistics of the collected views, dense model, proxy model and our algorithm on each scene are summarized in Table 1. # \mathcal{I} is the number of collected views. # V and # F are the number of vertices and faces in a 3D model, respectively. # P denotes the number of proxy polygons extracted from the proxy model.

3 ABLATION STUDY

We conduct ablation experiments to validate the effectiveness of our proposed view selection, image stitching and texture inpainting components, respectively.

3.1 View Selection

We first evaluate the influence of photometric and perspective terms in view selection. We propose two metrics to visualize the quality of input camera views $\{\mathcal{I}\}$ to a proxy polygon \mathcal{P}_j : i) Coverage. This is to measure the observation frequency of input views over \mathcal{P}_j . We subdivided the plane into small triangles and calculate the frequency

Table 1. Statistics on collected images and proxies of all the scenes. We also report the parameters we employed for texturing each example.

Scene	Views		Dense Model		Proxy Model		TwinTex Parameters		
	Name	# \mathcal{I}	Resolution	# \mathcal{V}	# \mathcal{F}	# \mathcal{V}	# \mathcal{F}	# \mathcal{P}	($\lambda_g, \lambda_c, \lambda_i, \lambda_{merge}, \lambda_d$)
Apartment		473	4864 × 3648	77,388	134,147	310	614	146	(0.95, 0.05, 10, 5, 10)
Factory		589	4864 × 3648	252,822	424,876	110	214	61	(0.95, 0.05, 50, 20, 10)
Polytech(teaser)		908	4864 × 3648	656,571	1,313,550	379	739	198	(0.95, 0.05, 50, 30, 50)
ArtSci		949	4864 × 3648	1,163,894	2,328,580	302	608	130	(0.95, 0.10, 10, 5, 10)
Hisense		1,710	4864 × 3648	526,756	964,530	206	406	113	(0.95, 0.05, 10, 5, 10)
Hitech		307	5472 × 3648	999,808	2,000,000	129	276	74	(1.20, 0.05, 30, 10, 20)
Bank		440	5472 × 3648	197,245	343,142	295	596	130	(0.95, 0.05, 10, 5, 10)
Hall		519	5472 × 3648	1,000,050	1,999,990	439	870	225	(0.95, 0.05, 10, 5, 10)
Headquarter		559	5472 × 3648	105,936	186,374	596	1,186	378	(0.90, 0.10, 10, 5, 10)
Highrise		828	5472 × 3648	362,605	724,003	33	62	18	(0.90, 0.10, 30, 10, 20)
Sunshine Plaza		945	5472 × 3648	1,499,228	3,000,000	1,325	2,613	733	(0.95, 0.05, 10, 5, 10)
Library		970	5472 × 3648	350,012	700,000	108	208	47	(2.00, 0.05, 10, 5, 10)
Center		1,049	5472 × 3648	360,984	626,906	139	272	65	(0.80, 0.20, 10, 5, 10)
Cabinet		220	5280 × 3956	103,393	204,880	64	119	32	(0.80, 0.20, 10, 5, 10)
Lab		594	5280 × 3956	677,060	1,072,651	53	93	25	(1.25, 0.05, 10, 5, 10)
Machine Room		1,234	5280 × 3956	289,235	362,980	118	211	59	(0.95, 0.10, 10, 5, 10)
CSSE		1,259	5280 × 3956	1,354,634	1,938,570	674	1,350	317	(1.00, 0.10, 10, 5, 10)
School		904	6000 × 4000	2,108,011	3,903,912	411	811	202	(0.95, 0.05, 10, 5, 10)

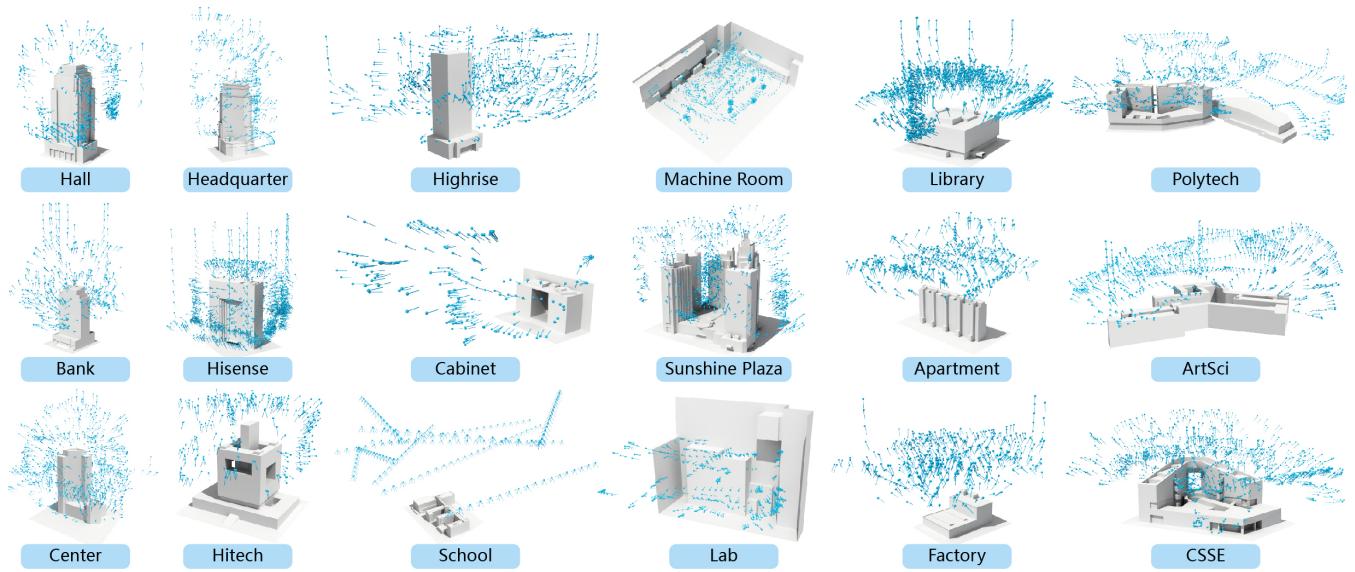


Fig. 2. The visualization of the path of an aerial drone (blue) collecting our input related to the proxy model (grey) for each scene.

of each triangle covering by \mathcal{R} . This value is normalized with the minimum and maximum values to better recognize the difference. We visualized the coverage from black to white (rarely observed to frequently observed). ii) Frontality. This is to measure the best level of front-parallel view in $\{\mathcal{I}\}$ for each pixel on \mathcal{P}_j . We first connect the camera center and the pixel position with a line. The frontality is then measured by the smallest angular difference between the

viewing direction of all cameras to the lines. We visualize this value from black to white (low to high frontality).

Two examples are shown in Fig. 3. Note the significant improvement of perspective consistency with the proposed perspective measurement (Q_{persp}) in both examples. Fig. 3 (a) shows the texturing results on a facade of the Center example with and without

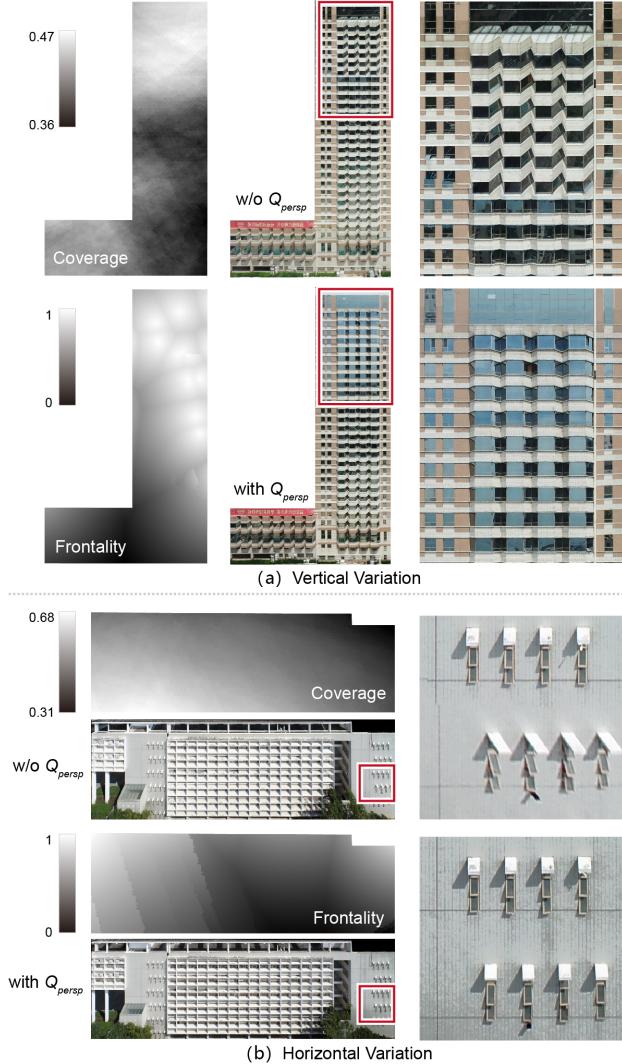


Fig. 3. Results of two examples with or without the perspective quality term on a proxy polygon. The coverage frequency and frontality of input views against a polygon are visualized from black (low) to white (high) as a color map.

perspective constraint. The selected textures suffer from large vertical perspective inconsistency of the bay windows without Q_{persp} , while with Q_{persp} these bay windows perfectly harmonize with each other. Fig. 3 (b) shows results on a facade of the ArtSci example with and without perspective constraint. We can observe a clear horizontal variation in the direction of the open windows in the result with only photometric constraints, while the directions fit well after considering the perspective quality.

By varying the weight on each term, we can trade off between photometric and perspective qualities. The larger weights on perspective and smoothness terms, the lower possibility to find a complete texture for a proxy polygon, i.e., the higher possibility to

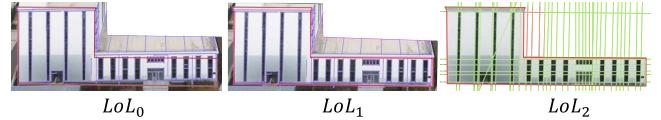


Fig. 4. Three levels of the line features extracted from a facade image of Factory example from local to global (left to right). The respective proxy boundary is visualized in red.



Fig. 5. Left: Texture maps with global and local misalignment before performing line-guided image stitching. Right: After performing image stitching with the adaptive mesh, we can maintain the alignment of line features.

generate missing parts. In our results, we found $\lambda_{smooth} = 0.05$ and $\lambda_{persp} = 0.2$ provide a good compromise.

3.2 Image Stitching

We report LoLs and registration results on two examples. Fig. 4 is a facade consisting of only one selected image on the Factory example. The first and second facade images show that the ICP algorithm can effectively register the texture images guided with LoL_0 . Fig. 9 shows a facade consisting of three selected partial images on Highrise example, indicating that our tool performs well with partial images. In both examples, the LoL_2 which is extracted based on corresponding LoL_1 depicts the structural layout well.

Next, we conduct an ablation study to validate the effectiveness of our designed adaptive mesh data structure for preserving line structures. To this end, in the image stitching stage, instead of using an adaptive mesh to warp the texture patch, we directly blend all the texture patches one by one for each plane. Fig. 5 shows the stitching results without and with adaptive mesh deformation on global border and local feature alignment. The comparison demonstrates that by using the adaptive mesh to locally deform the image, we are able to explicitly maintain the alignment of line segments in the overlapping areas of different texture patches.

3.3 Texture optimization

Illumination Adjustment. We now demonstrate the result of illumination adjustment on Hisense example in Fig. 6. Eight photos are selected from 1,710 input images, from which we can see that the projected photos have a large variance on illumination condition. The



Fig. 6. Resultant harmonic texture map of the large ground in Hisense example after performing illumination adjustment with eight selected images. The red box indicates the reference image.

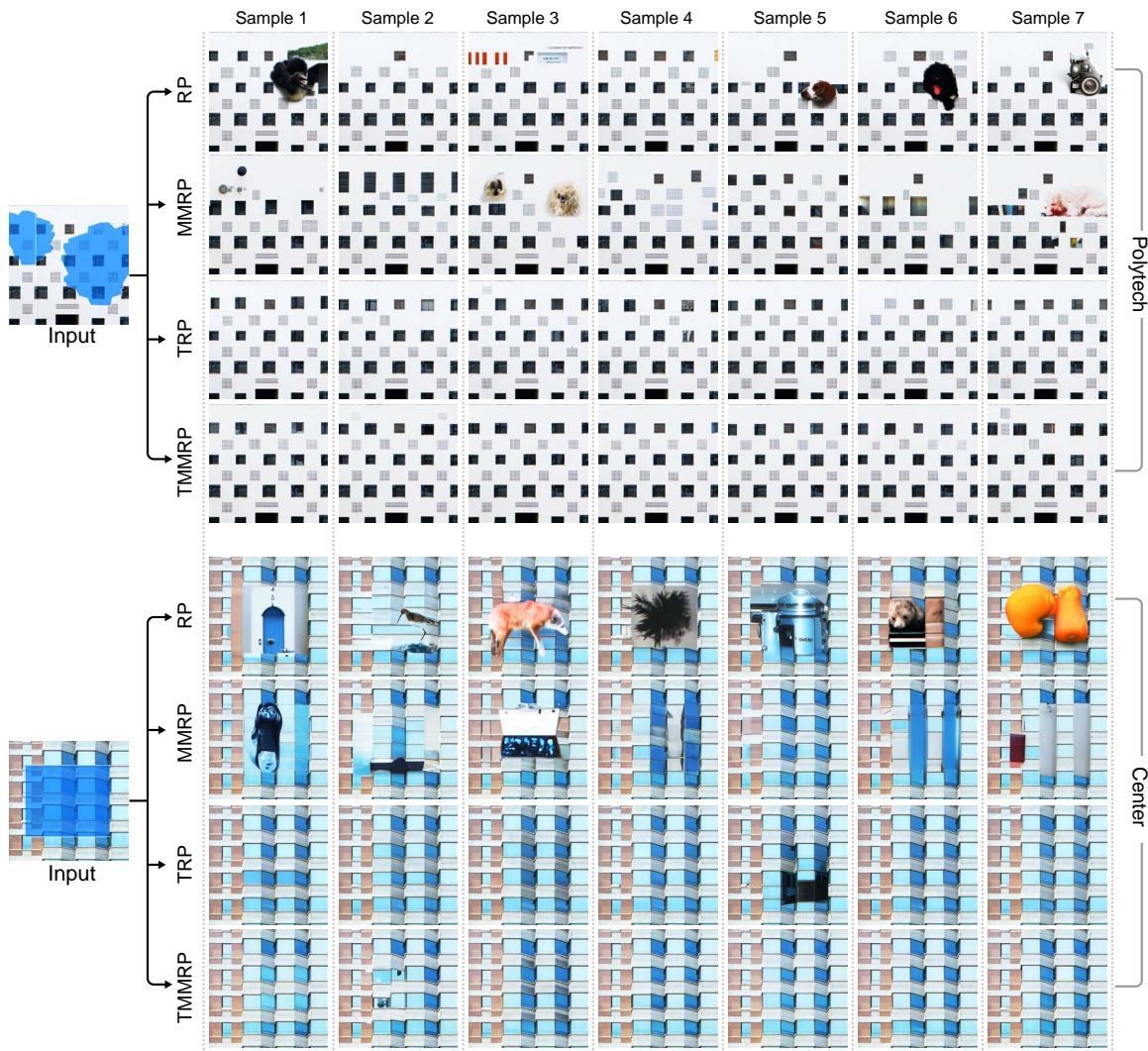


Fig. 7. Inpainting results generated with RP, MMRP, fine-tuned RP (TRP), fine-tuned MMRP (TMMRP), respectively.

Table 2. Quantitative comparison on inpainting performance.

Scene	Method	$SSIM \uparrow$	$PNSR \uparrow$	$LPIPS \downarrow$
Polytech	RP	0.8575	18.3306	0.1693
	MMRP	0.8500	18.4028	0.2335
	TRP	0.8935	22.0736	0.0924
	TMMRP	0.8869	21.5704	0.0848
Center	RP	0.7295	17.1720	0.2515
	MMRP	0.7957	19.3460	0.1966
	TRP	0.7840	22.0645	0.0729
	TMMRP	0.7976	23.2798	0.0659

illumination of each projected photo is then adjusted towards the reference image which has the largest projection area. Obviously, the enlarged texture after adjustment exhibits high photometric quality.

Image Inpainting. There may be possible missing regions that have not been covered by any input photos after previous steps. Lastly, we show the improvement of our inpainting scheme in generating coherent content. This step is to predict missing pixels of an image using a mask region as a condition. Lugmayr et al. [2022] used a trained unconditional denoising diffusion probabilistic model for the inpainting task.

Given an image with missing regions, we generated seven inpainted textures using original RePaint (RP) [Lugmayr et al. 2022], the fine-tuned RP with our TwinTexSet (TRP), our MMRP and fine-tuned MMRP (TMMRP), respectively. We report three commonly reported perceptual metrics: SSIM, PNSR and LPIPS [Kastrayulin et al. 2022]. The average value of these metrics over each set is calculated on two examples. Quantitative results are shown in Table 2, and Fig. 7 shows the visual comparison. The results generated with RP are geometrically consistent with high photometric quality. However, there appears plenty of irrelevant objects in the inpainted regions. This is greatly improved with our MMRP. But limited by the training data, the MMRP still cannot solve this issue. Our TMMRP further improves the geometric and semantic quality as shown in the results.

3.4 Stage Results

Fig. 9 shows the overview of our TwinTex on a facade of Highrise building. We effectively selected 3 high-quality images from 828 input photos and generated a complete texture map for the target proxy polygon.

3.5 Stress Test

Fig. 10 shows a stress test and the stage result on Factory example. We randomly removed 50%, 75% and 95% photos from the original set and feed the remaining views to our TwinTex. The textured proxies after view selection are generated by simply overlapping the projection of selected images following the reverse order of selection. In the bottom row, we only use 29 images to texture this example with 61 facades (each plane has about 5 photos to select from). The result on the second column shows that our tool effectively selected

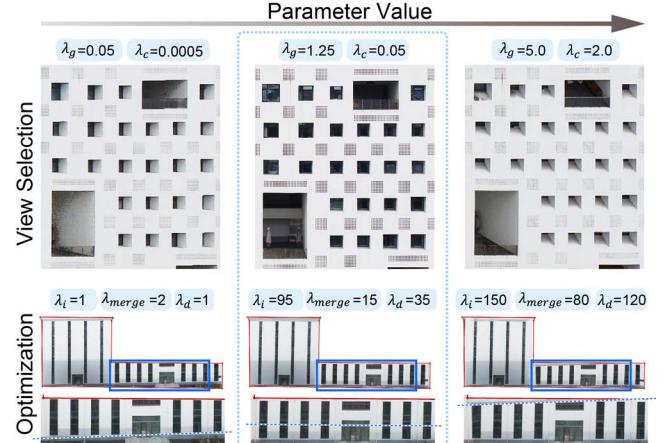


Fig. 8. Results of view selection (top) and image optimization (bottom) with various sets of parameters. The results inside the blue dash rectangle denote the generated results with parameters inside the recommended ranges.

photos covering the model under extremely limited resources. The result on the third column shows that our stitching methodology successfully fixed misalignment among linear features. The result on the last column shows that our inpainting approach is able to infill missing regions with consistent pixels.

4 PARAMETER SETTINGS

In Table 1, we provide the values of main parameters used for texturing each example in this paper. We also list the recommended ranges on the value of each parameter in Table 3 for the reference of users. The parameter values that we mostly use are set as default in our Plug-in.

We also report several non-ideal results using parameters out of the recommended range of values in Fig. 8. For view selection on the Polytech example (top), the set of parameters with very small values causes results with lower resolution and extremely inclined angle, while extremely large values generate results with high resolution and less confronting angle. For the optimization of the facade on Factory examples (bottom), the set of parameters with very small values generates not sufficiently registered result while extremely large values lead to overly registered result. Although the results are acceptable, we suggest our users to adopt the parameter values in the recommended range for the best performance of TwinTex.

5 ADDITIONAL COMPARISON

5.1 Image Stitching

Moreover, we also compare with two state-of-the-art image stitching approaches on Polytech, Highrise and Hitech, including [Liao and Li 2019] and [Jia et al. 2021], both of which apply a uniform grid to warp the images to maintain the local geometry features. In order to quantify the stitching performance on linear structures, we adopt the metrics introduced by [Jia et al. 2021] to measure the distance of matched lines and the discrepancy in the direction of matched lines. Specifically, the distance term E_{dis} represents the average distance

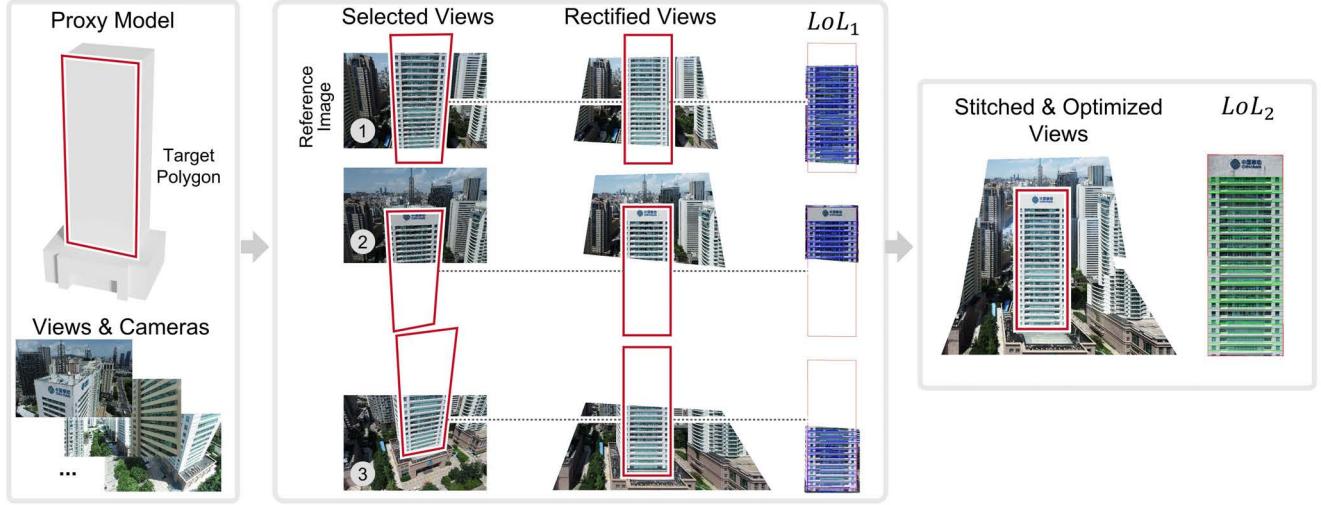


Fig. 9. Overview of our TwinTex on a facade of the Highrise building.

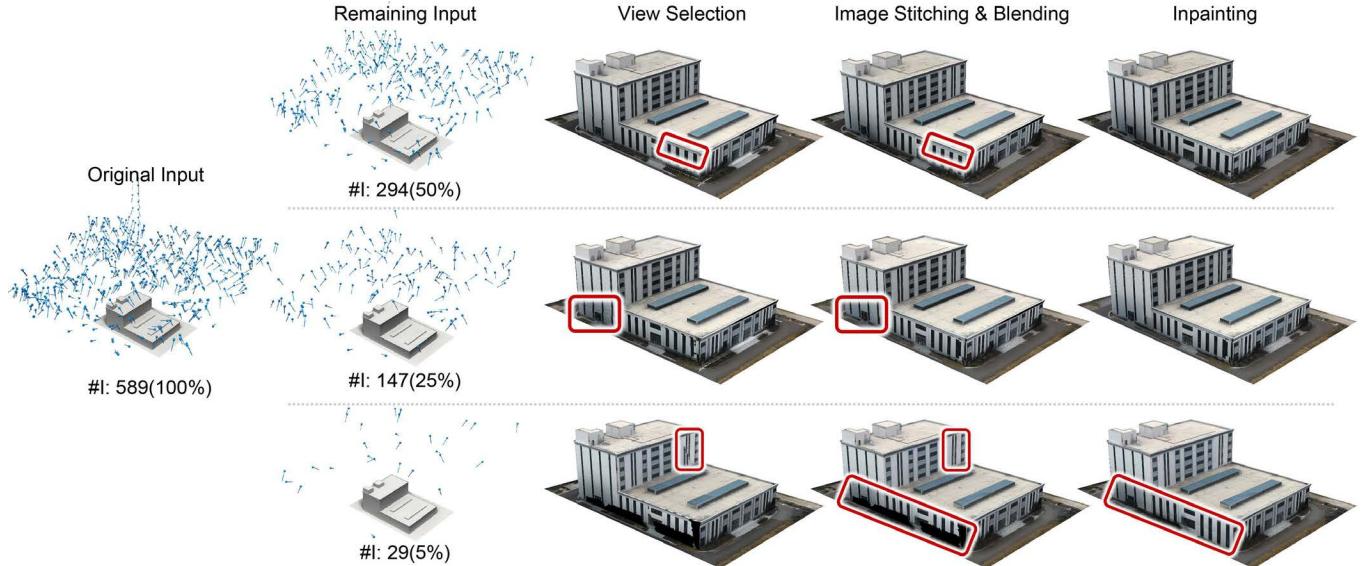


Fig. 10. Stress test of our methodology on Factory example via randomly removing a certain percentage of photos from original input. In each row we visualize: the camera distribution of the remaining input, the textured proxy based in remaining input after view selection, image stitching with blending, and inpainting. #I denotes the number of photos. The value in the bracket denotes the percentage of the remaining photos related to original input.

between two warped endpoints and the matching line pairs, which is described as:

$$E_{dis} = \sqrt{\frac{1}{L} \sum_{j=1}^{i=L} \left\| \frac{dis(l'_j, p_s^j) + dis(l'_j, p_e^j)}{2} \right\|^2}, \quad (1)$$

where l'_j represents the j^{th} matching line of the reference image, p_s^j and p_e^j are two endpoints of the j^{th} matching line in target image. The direction term E_{dir} computes the direction difference between

the warped line and the matching line, which is defined as:

$$E_{dir} = \sqrt{\frac{1}{L} \sum_{j=1}^{i=L} \|sin(\theta)\|^2}, \quad (2)$$

where θ denotes the angle between the warped line and the matching line.

From three scenes we choose a set of representative planes with high-quality textures and a large number of matching lines for quantitative comparison of different stitching methods. Table 4

Table 3. The recommended range of values on each parameter for simply tuning.

Parameter	Function	Suggestion
λ_c	Larger value prefers image with higher photometric consistency.	0.02 ~ 0.20
λ_g	Larger value prefers image with higher resolution.	0.80 ~ 2.00
λ_i	Larger value to find ICP matching points within a larger range of neighboring boundary points.	10 ~ 100
λ_{merge}	Increase this value to merge the end-points of line segments within a larger range of neighboring boundary points.	5 ~ 30
λ_d	Increase this value to match line segments in a larger range.	10 ~ 50

Table 4. Quantitative comparison on stitching performance of different image stitching methods.

Scene	Method	$E_{dis} \downarrow$	$E_{dir} \downarrow$
Highrise	[Liao and Li 2019]	0.9294	0.0059
	[Jia et al. 2021]	1.7888	0.0162
	Ours	0.9006	0.0024
Hitech	[Liao and Li 2019]	2.9038	0.0121
	[Jia et al. 2021]	2.6385	0.0167
	Ours	2.4345	0.0048
Polytech	[Liao and Li 2019]	2.0011	0.0117
	[Jia et al. 2021]	1.5922	0.0127
	Ours	2.0597	0.0103

reports the performance for the preservation of linear structures using the above two measures. From the comparison, we can see that our method outperforms [Liao and Li 2019] and [Jia et al. 2021] in distance and differences of angles. This improvement mainly comes from the property of our adaptive mesh which takes geometric primitives as constraints and controls the deformation of each line segment independently comparing against the grid mesh involved in [Liao and Li 2019] and [Jia et al. 2021].

5.2 Texture Inpainting

Fig. 11 shows our inpainting results comparing to [Huang et al. 2014] which also performs image completion using planar structure guidance (CPSG). It indicates that our MMRP result exhibits a higher level of semantic and geometric consistency.

5.3 Texturing Methods

In this subsection, we evaluate the quality of our generated texture maps by comparing them against RealityCapture (RC) and two competitive texturing generation frameworks, let there be color (LTBC) [Waechter et al. 2014] and patch based optimization (PBO) [Bi et al. 2017].

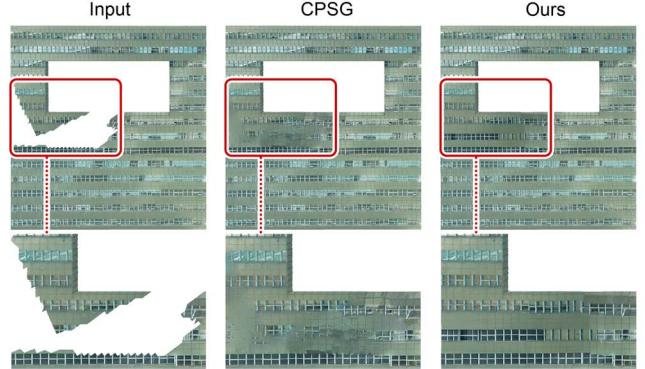


Fig. 11. Inpainting results of CPSG [Huang et al. 2014] and our MMRP on a facade of Hitech with an irregular missing region.

Table 5. Quantitative comparison on the results in Fig. 12 and Fig. 13.

Scene	Method	SSIM \uparrow	LPIPS \downarrow	Time (min)
Hitech	RC	0.5738	0.4938	141.5
	LTBC	0.5989	0.4465	1.9
	Ours	0.6251	0.4424	100.0
Center	RC	0.350	0.831	287.0
	LTBC	0.354	0.836	4.4
	Ours	0.358	0.831	167.4
Artsci	RC	0.346	0.883	320.9
	LTBC	0.331	0.873	6.0
	Ours	0.348	0.877	256.1

We adopt the evaluation scheme proposed by Waechter et al. [2017] to compare the rendered image with the corresponding real image following a specific view from the input cameras. We select the ground truth photos for evaluation from the input excluding the photos for texturing. Two visual similarity metrics, namely the structural similarity index measure (SSIM) and learned perceptual image patch similarity (LPIPS) [Kastryulin et al. 2022], are adopted for quantitative evaluation.

We first performed an extra comparison on three example against RC and LTBC. The visual comparisons are shown in Fig. 12 and Fig. 13. Fig. 13 shows the comparison on full textured models of our approach, RC and LTBC. It still indicates that our results achieve a higher level of completeness, geometric consistency and photometric quality. The quantitative comparison results are given in Table 5.

After that, we locally perform the comparison to PBO using four planes per example. To this end, we fed PBO with the set of views selected by our algorithm for the planar polygons involved in the comparison. Fig. 14-15 shows the texturing results of RC, LTBC, PBO and ours on six examples. Quantitative results are listed in Table 6. Since SSIM is by nature less sensitive to blurring issues and possibly gives higher scores to images with such artifacts [Zhang et al. 2018], the texture with the regional blurring problem could have higher scores. Meanwhile, LPIPS is designed to match human



Fig. 12. Visual comparison against RC, LTBC and ours on three examples. The detailed comparisons are shown in the zoomed-in insets.

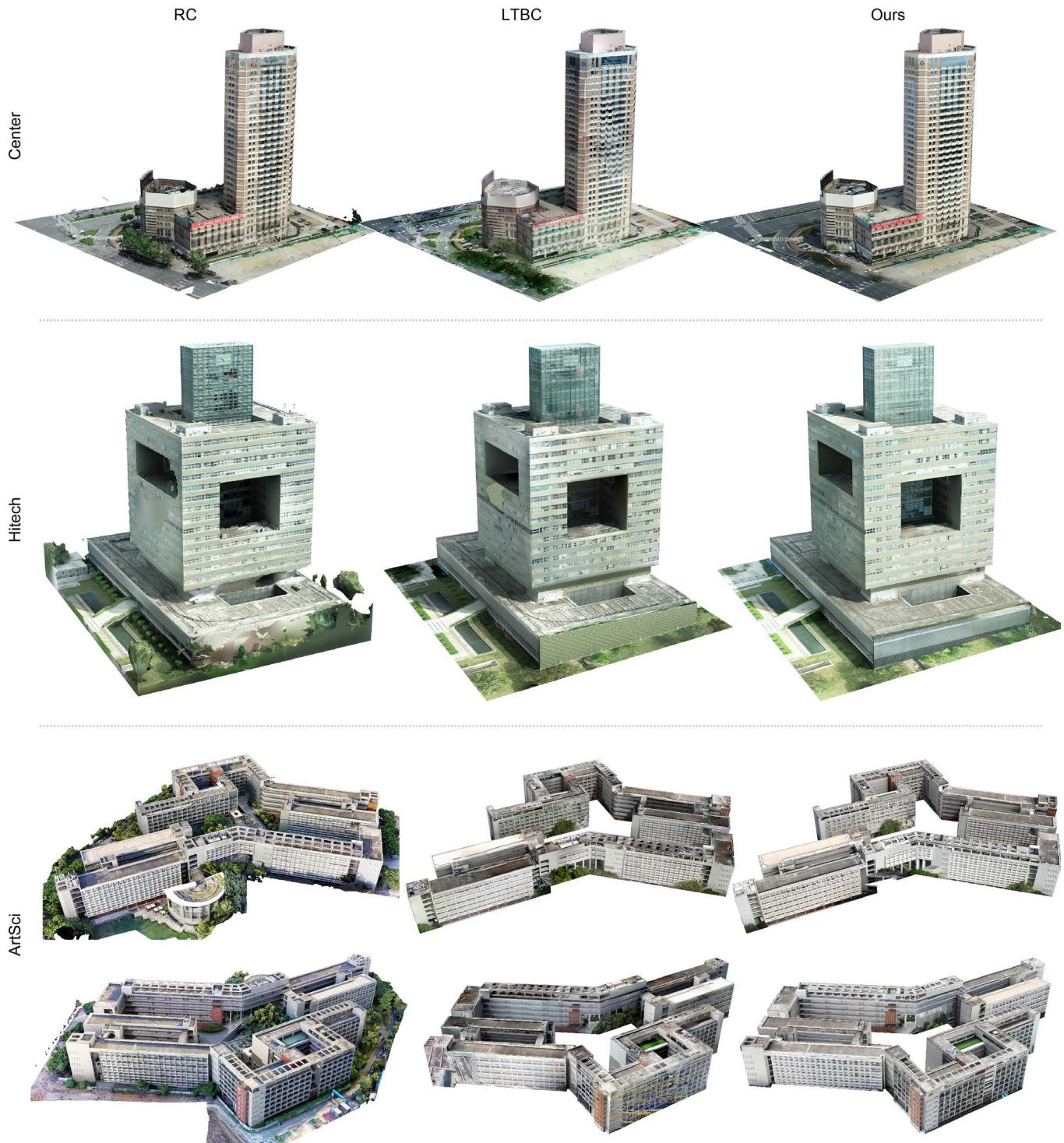


Fig. 13. Visual comparison of our overall views against RC and LTBC on three more examples.



Fig. 14. Visual comparison against RC, LTBC, PBO and ours on three examples. The detailed comparisons are shown in the zoomed-in insets.



Fig. 15. Visual comparison against RC, LTBC, PBO and ours on three examples. The detailed comparisons are shown in the zoomed-in insets.

Table 6. Quantitative comparison on the performance of texturing results. Numbers in the bracket denote the quality value of the four zoomed-in views in Fig. 14 and Fig. 15 from top to bottom.

Scene	Method	SSIM ↑	LPIPS ↓	Time (min)
Highrise	RC	{0.2762, 0.1607, 0.3408, 0.2945}	{0.5716, 0.6480, 0.6320, 0.6206}	123.00
	LTBC	{0.2854, 0.1666, 0.3455, 0.3418}	{0.4889, 0.6363, 0.6502, 0.5217}	9.46
	PBO	{0.3115, 0.2041, 0.3918, 0.3679}	{0.5863, 0.6841, 0.6614, 0.6206}	758.26
	Ours	{0.2589, 0.1874, 0.4307, 0.3091}	{0.5771, 0.6359, 0.6319, 0.2653}	34.72
Hisense	RC	{0.0854, 0.1325, 0.0840, 0.3138}	{0.3379, 0.3192, 0.4550, 0.6634}	792.32
	LTBC	{0.2908, 0.1527, 0.0812, 0.3357}	{0.3868, 0.4333, 0.4811, 0.7139}	12.04
	PBO	{0.2988, 0.1663, 0.1165, 0.5236}	{0.4309, 0.3672, 0.4324, 0.4466}	1054.63
	Ours	{0.2932, 0.2339, 0.0781, 0.5327}	{0.1203, 0.1032, 0.3026, 0.2393}	438.82
School	RC	{0.1777, 0.2517, 0.3847, 0.5782}	{0.5837, 0.4434, 0.5512, 0.5941}	375.11
	LTBC	{0.2173, 0.1857, 0.4011, 0.6426}	{0.5367, 0.5240, 0.6590, 0.5252}	4.80
	PBO	{0.4055, 0.4816, 0.3972, 0.7445}	{0.7204, 0.5273, 0.5758, 0.3940}	893.60
	Ours	{0.2123, 0.5947, 0.4194, 0.6952}	{0.5160, 0.2518, 0.5150, 0.2888}	284.20
Library	RC	{0.2763, 0.2758, 0.3223, 0.5297}	{0.5247, 0.5981, 0.6814, 0.6496}	383.59
	LTBC	{0.2931, 0.2869, 0.3442, 0.4553}	{0.5377, 0.6811, 0.6770, 0.4806}	4.84
	PBO	{0.6261, 0.4508, 0.3828, 0.7402}	{0.6250, 0.6996, 0.6907, 0.8346}	810.73
	Ours	{0.4625, 0.3808, 0.3635, 0.4641}	{0.2806, 0.5180, 0.6595, 0.4670}	200.44
Bank	RC	{0.6085, 0.2484, 0.6184, 0.2610}	{0.2804, 0.4257, 0.2309, 0.5348}	198.26
	LTBC	{0.6700, 0.2006, 0.5231, 0.3302}	{0.1912, 0.5071, 0.3740, 0.5324}	6.41
	PBO	{0.6463, 0.2461, 0.6952, 0.2659}	{0.2157, 0.4311, 0.3050, 0.5740}	672.10
	Ours	{0.9650, 0.2598, 0.6246, 0.2752}	{0.0241, 0.2714, 0.1816, 0.5229}	174.51
Apartment	RC	{0.2602, 0.5340, 0.3361, 0.4225}	{0.3526, 0.3618, 0.4103, 0.5158}	205.76
	LTBC	{0.2305, 0.4461, 0.3049, 0.4211}	{0.3813, 0.4659, 0.4808, 0.6129}	3.00
	PBO	{0.4395, 0.5333, 0.4471, 0.6340}	{0.3332, 0.3884, 0.6198, 0.4811}	806.00
	Ours	{0.3465, 0.6718, 0.4384, 0.5780}	{0.1072, 0.1516, 0.2136, 0.3436}	185.42



Fig. 16. Visual comparison against manual work and ours on three LODs.



Fig. 17. Visual comparison of two challenging cases against RC, LTBC and ours. The detailed comparisons are shown as zoomed-in insets.

perception and yield better scores for images with higher level of coherence. Note that the recorded time cost for PBO does not include view selection or is not for texturing the entire proxy although it is relatively long.

5.4 Levels of Abstraction

We evaluate the quality of textures generated with manual work and our method, which share the same plane-oriented logic. This experiment is performed on Polytech with three levels of details from coarse to fine: LOD0 ($V : 60$, $\mathcal{F} : 97$, $\mathcal{P} : 26$), LOD1 ($V : 171$, $\mathcal{F} : 222$, $\mathcal{P} : 75$) and LOD2 ($V : 379$, $\mathcal{F} : 739$, $\mathcal{P} : 198$). The manually textured results are created by a professional modeler. These results are visualized in Fig. 16. We can see that both methods yielded vivid texture. However, with the increasing level of details, the times needed to create such high quality textures via manual work grows from days to more than a week. The significant amount of time consuming and labor intensity make manual texturing less

feasible in applications for a complex proxy or a larger-scale urban scene. Meanwhile, our tool can automatically align texture to geometry, preserve line-features and finally produce texture maps with competitive quality using much less time.

5.5 Challenging Cases

Finally, we conduct experiments on two challenging urban scenes that contain curve surfaces, facades with non-linear features, and facades with a lot of reflection, as shown in Fig. 17.

The visual comparisons further suggest that the textures generated by our approach have several significant advantages over previous methods: i) closer to real photos with perspective consistency and harmonic illumination, ii) contain rarely border misalignment, distortion, blurring and seaming artifacts resulting from inaccurate camera parameters, iii) contain rarely ghosting effects resulting from dynamic instances, iv) preserve geometric details better, and v) fill

the missing regions with geometrically and semantically consistent contents.

ACKNOWLEDGMENTS

We thank the reviewers for their constructive comments. This work was supported in parts by NSFC (U21B2023, U2001206, U22B2034, 62172416, 62302313), DEGP Innovation Team (2022KCXTD025), and Shenzhen Science and Technology Program (KQTD202108110900440 03, RCJC20200714114435012, JCYJ20210324120213036).

REFERENCES

- Sai Bi, Nima Khademi Kalantari, and Ravi Ramamoorthi. 2017. Patch-based optimization for image-based texture mapping. *ACM Trans. Graph. (SIGGRAPH)* 36, 4 (2017), 106:1–106:11.
- Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. 2014. Image completion using planar structure guidance. *ACM Trans. Graph. (SIGGRAPH)* 33, 4 (2014), 129:1–129:10.
- Qi Jia, ZhengJun Li, Xin Fan, Haotian Zhao, Shiyu Teng, Xinchen Ye, and Longin Jan Latecki. 2021. Leveraging line-point consistence to preserve structures for wide parallax image stitching. In *IEEE Computer Vision and Pattern Recognition (CVPR)*.
- 12186–12195.
- Sergey Kastrulyin, Jamil Zakirov, Denis Prokopenko, and Dmitry V Dylov. 2022. PyTorch Image Quality: Metrics for Image Quality Assessment. *arXiv preprint arXiv:2208.14818* (2022).
- Tianli Liao and Nan Li. 2019. Single-perspective warps in natural image stitching. *IEEE Trans. Image Process.* 29 (2019), 724–735.
- Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. 2022. Repaint: Inpainting using denoising diffusion probabilistic models. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 11461–11471.
- Michael Waechter, Mate Beljan, Simon Fuhrmann, Nils Moehrle, Johannes Kopf, and Michael Goesele. 2017. Virtual Rephotography: Novel View Prediction Error for 3D Reconstruction. *ACM Trans. Graph.* 36, 1, Article 8 (2017), 11 pages.
- Michael Waechter, Nils Moehrle, and Michael Goesele. 2014. Let there be color! Large-scale texturing of 3D reconstructions. In *European Conference on Computer Vision (ECCV)*, 836–850.
- Han Zhang, Yucong Yao, Ke Xie, Chi-Wing Fu, Hao Zhang, and Hui Huang. 2021. Continuous Aerial Path Planning for 3D Urban Scene Reconstruction. *ACM Trans. Graph. (SIGGRAPH Asia)* 40, 6 (2021), 225:1–225:15.
- Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 586–595.
- Xiaohui Zhou, Ke Xie, Kai Huang, Yili Liu, Yang Zhou, Minglun Gong, and Hui Huang. 2020. Offsite Aerial Path Planning for Efficient Urban Scene Reconstruction. *ACM Trans. Graph. (SIGGRAPH Asia)* 39, 6 (2020), 192:1–192:16.