Inpainting Applied to Facade Images: a Comparison of Algorithms

Willy Fritzsche, Steffen Goebbels[0000−0003−4313−9101] ✉, Simon Hensel, Marco Rußinski, and Nils Schuch

Niederrhein University of Applied Sciences, Faculty of Electrical Engineering and Computer Science, iPattern Institute, 47805 Krefeld, Germany
{willy.fritzsche, marco.russinski, nils.schuch}@stud.hn.de,
{steffen.goebbels, simon.hensel}@hsnr.de

This is a self-archived version of the paper published in: M. El Yacoubi et al: Pattern Recognition and Artificial Intelligence, Proc. ICPRAI 2022 Part I, LNCS 13363, Springer, Berlin, pp. 410–422, 2022,
https://doi.org/10.1007/978-3-031-09037-0_34

Abstract. Many municipalities provide textured 3D city models for planning and simulation purposes. Usually, the textures are automatically taken from oblique aerial images. In these images, walls may be occluded by building parts, vegetation and other objects such as cars, traffic signs, etc. To obtain high quality models, these objects have to be segmented and then removed from facade textures. In this study, we investigate the ability of different non-specialized inpainting algorithms to continue facade patterns in occluded facade areas. To this end, non-occluded facade textures of a 3D city model are equipped with various masks simulating occlusions. Then, the performance of the algorithms is evaluated by comparing their results with the original images. In particular, very useful results are obtained with the neural network "DeepFill v2" trained with transfer learning on freely available facade datasets and the "Shift-Map" algorithm.

Keywords: Inpainting · 3D City Models · Facade Textures

1 Introduction

During the the last decades, various image inpainting techniques have been developed that fill absent regions in images. In [15], an extensive survey is given. Inpainting is an ill-assorted problem because there is no hard criterion how missing information has to look like. However, generated edges and texture patterns should somehow fit with the given data. Such reconstruction of missing information is often required for optical remote sensing data, see the overview paper [19]. Our goal is to restore areas on facade images that are occluded, e.g., by vegetation or other buildings. These facade images are used as textures in 3D city models, cf. Figure 1, and they are typically obtained from oblique aerial images.

Occluded facade regions can be segmented by detecting objects like trees and vehicles as well as by drawing the 3D model from the camera perspective. The 3D model then helps to identify regions hidden by other buildings, cf. [7]. There exist specialized inpainting algorithms for building facades, see Section 2. However, the contribution of this paper is to investigate whether readily available general methods can be used instead of specialized algorithms. The occluded regions are usually so large that realistic results are difficult to obtain by merely considering only local data from the region boundary. Nevertheless, we consider the boundary-based Navier-Stokes and the Telea diffusion algorithms [1, 21] (see Section 3.1) to compare the results with global inpainting, i.e., with texture synthesis approaches. Facade images often show a repetitive pattern due to a regular arrangement of windows. Such patterns can be described by grammars and they are the reason why the continuation of global texture properties can work sufficiently for facades. We compare the results of both explicitly implemented algorithms and deep neural networks, see Section 3.3 for the two convolution based deep neural networks DeepFill v2 [27] and GMCNN [23] as well as Section 3.2 for other algorithms.

It turns out that global texture synthesis works better than local inpainting algorithms. Due to technical problems, Frequency Selective Reconstruction [5, 18] operates only very well on downscaled images. Without adjusting the scale, the best results are gained with transfer learning applied to DeepFill v2 and the example-based Shift-Map inpainting algorithm, see Section 5.

2 Related Work

Whereas we discuss the application of general purpose algorithms, various papers have treated the specialization of inpainting techniques to facade images. A selection of these algorithms is briefly described in this section.

Facades of large buildings often have a regular pattern of windows that can be expressed with an Irregular Rectangular Lattice (IRL). This is a grid of lines that extend the boundaries of semantically labeled facade objects like windows and doors. Based on an IRL, a recurrent neural network was used in [9] to propose positions and sizes of occluded windows. In the context of 3D city models, the algorithm in [3] synthesizes photorealistic facade images using example-based inpainting. To this end, textured tiles are defined by the rectangles of the IRL obtained from a random forest. The IRL is then extrapolated to occluded regions. A genetic algorithm is applied to optimize a labeling of the rectangular tiles. The algorithm decides which rectangles show the same textures. These textures are taken from non-occluded rectangle instances. Such tile based synthesis was also discussed and applied to facades in [25]. In [13], information about detected facade objects was combined with example-based inpainting, and in [28], instance segmentation of unwanted objects was combined with a generative adversarial network (GAN) to fill regions occupied by the objects.

The EdgeConnect GAN [16] was slightly improved for facade images in [14] by applying semantic segmentation. It was extended to a three-stage model that uses three GANs to reconstruct an edge image, a label image, and a texture image.

Another approach to reconstruct missing facade regions is to apply split grammars, see [24]. These grammars are a collection of rules that describe the placement of facade objects. With their help, missing facade objects can be added and even complete facades can be generated procedurally. Often, facades show symmetry. This is utilized in the algorithm [2].

Facade regions can be also reconstructed with the inpainting algorithm in [10]. After detecting line segments of edges, the segments are classified according to vanishing points of corresponding lines. Then image regions covered by line segments that belong to two classes are likely to describe a plane in 3D. Textures are continued by considering these planes.

Often, facades are visible in multiple oblique aerial images belonging to different camera positions and directions. Algorithms like the one in [12] address facade texture generation based on images taken from a moving camera.

3 Algorithms

For this study, we have selected algorithms that are freely available via the computer vision library OpenCV[1] or via open code repositories.

3.1 Local, Diffusion-based Inpainting Algorithms

OpenCV offers traditional inpainting algorithms that locally continue patterns from the boundary of a region to its interior. They require a source image and a separate 8-bit, one channel mask to define occluded areas. These algorithms are also classified as structure-based.

Navier-Stokes (NS) algorithm The Navier-Stokes equations are partial differential equations that model the motion of viscous fluids. Applied to image inpainting, the equations can be used to continue isophotes (curves connecting points of equal gray value) while matching gradient vectors at the boundary of inpainted regions, see [1]. Due to the diffusion process, some blur is visible if the algorithm is applied to fill a larger region.

Inpainting based on the Fast Marching Method (Telea algorithm) In [21], Alexandru Telea presented an inpainting algorithm that is easier to implement than the NS algorithm. It iteratively propagates boundary information to the interior of occluded regions by computing weighted means of pixel values that are estimated with a linear Taylor approximation. Thus, as with many algorithms, the region is synthesized from the outside inward.

[1] `https://opencv.org` (all websites accessed on January 12, 2022)

3.2 Global Inpainting Algorithms Not Relying on Deep Learning

With the Shift-Map and the Frequency Selective Reconstruction methods, we take into account two algorithms that are provided by the “xphoto” package[2] of OpenCV. These algorithms do not only consider the boundary of occluded regions but the whole image. They are also called texture-based.

Shift-Map algorithm A shift-map consists of offsets that describe how pixels are moved (shifted, transformed) from a source to a target image region. Shift maps can be optimized with respect to certain smoothness and consistency requirements by solving a graph labeling problem, see [17]. The cited paper also introduced shift-map-based inpainting: By choosing an occluded area as a source, inpainting can be viewed as finding an optimized shift-map. The xphoto-implementation is based on [8] where a sparsely distributed statistics of patch offsets was utilized to implement example-based inpainting. This algorithm can be seen as a generalized variant of example-based synthesis of facade patterns proposed with algorithms mentioned in Section 2.

Frequency Selective Reconstruction (FSR) In contrast to local inpainting techniques based on boundary information, Fourier analysis is a means for global approximation due to the global support of basis functions. When Fourier coefficients are determined based on known image areas, the Fourier partial sums also provide data for unknown areas. This is the idea behind Frequency Selective Inpainting, see [11], which is based on the discrete Fourier transform. Discrete Fourier coefficients are estimated from the given, incomplete sample data. The coefficients can be seen as factors in a linear combination of Fourier basis functions to represent the given discrete data. Since the given information is incomplete, the corresponding system of linear equations is underdetermined, and there are infinitely many solutions for the coefficients. Therefore, the method applies a heuristics called Matching Pursuit. It iteratively selects a basis function that best approximates the given data. In each iteration, this best approximation is subtracted from the given data (residual vector). Thus, rather than calculating all the coefficients at once, iterations are performed by selecting the most important frequencies. Once the discrete coefficients are estimated, the data can be reconstructed by the inverse discrete Fourier transform. The xphoto-implementation follows [5] and [18].

3.3 Deep Learning-based Global Inpainting Algorithms

DeepFill v2 The ”Free-Form Image Inpainting with Gated Convolution” network, “DeepFill v2” for short, is based on gated convolution, see [26, 27]. In contrast to the application of partial convolution, gated convolution allows the network to learn how to apply convolution kernels to incomplete data. While

[2] `https://docs.opencv.org/5.x/de/daa/group__xphoto.html`

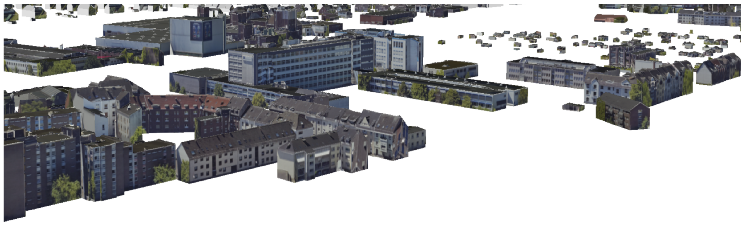


Fig. 1. Textured city model of Krefeld

the features are based on general convolution, the algorithm uses an adaptive dynamic feature selection mechanism (known as gating mechanism) for each channel at each spatial position. When applied, the network consists of two separate encoder-decoder sub-networks, the coarse network and the refinement network which implements contextual attention. An input mask defines the regions to be filled. In these regions, the output of the coarse network looks like a blurred image. The contextual attention stage adds the missing details such as contours. In the training phase, a third sub-network is attached to compute an adversarial loss that is linearly combined with a pixel-wise l_1 reconstruction loss.

GMCNN The Generative Multi-column Convolutional Neural Network (GMCNN) uses local as well as global information to predict pixels in regions that are specified by a mask, see [23]. In total, the network consists of three sub-networks. The inpainting is done with the first sub-network, the generator. The second subnetwork implements local and global discriminators for adversarial training and the third sub-network is a pre-trained VGG network [20] that provides features to calculate the implicit diversified Markov random fields (ID-MRF) loss introduced in [23]. With respect to the feature space, this loss minimizes the distance between the generator output and a nearest neighbor in the set of ground truth images. Only the first sub-network is used for testing. This generator consists of three parallel encoder-decoders, which help to determine features on multiple scales, and a shared decoder module to reconstruct the image.

4 Ground Truth, Training Data, and Network Training

Our aim is to improve facade textures obtained from oblique aerial images. Thus, we generated a realistic set of test images from a textured 3D city model of the area around our institute. We previously computed this model with the method described in [6] based on airborne laser scanning point clouds and cadastral footprints (available from the state cadastral office of the German state North Rhine-Westphalia[3]) and textured it with oblique aerial images provided by the

[3] https://www.bezreg-koeln.nrw.de/brk_internet/geobasis/hoehenmodelle/3d-messdaten/index.html

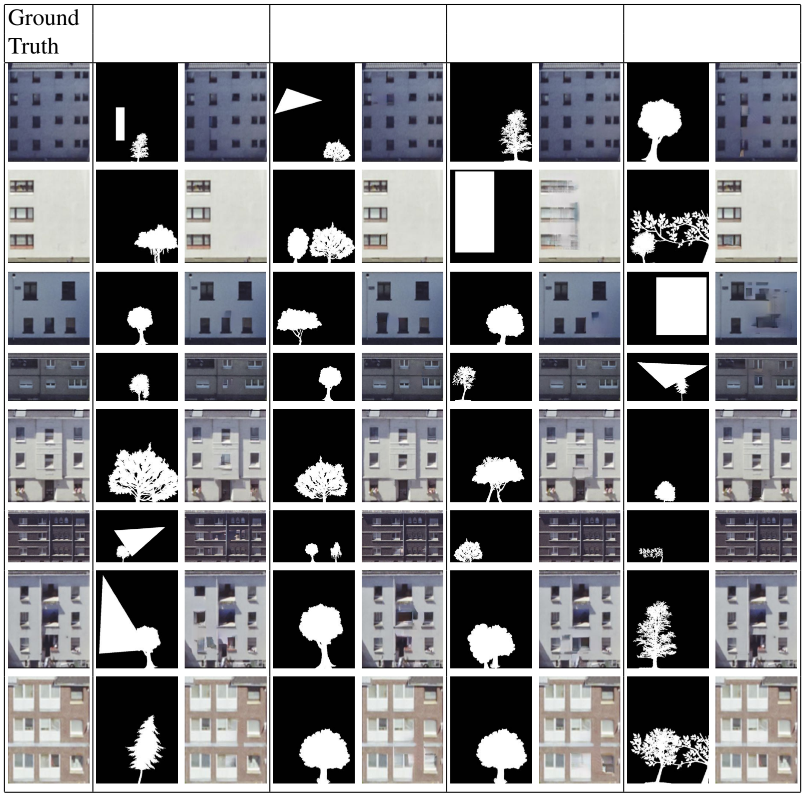


Fig. 2. Results of the DeepFill v2 scenario (S2) with regard to different masks

city of Krefeld, see Figure 1. The obtained facade images had a resolution of about 15 × 15 pixels per square meter and were free of perspective distortions. For this study, we considered only rectangular images that were completely covered with facades and did not show a background like the sky. To define a ground truth, we manually selected 120 images that were free of occlusions. This proved to be no easy task since most facade textures showed occlusions (mostly vegetation) of different sizes. We increased the number of images to 206 by mirroring and periodic repetition. Together with automatically generated occlusion masks, we tested all algorithms on these images. These masks were assembled from filled triangles, rectangles, circles and higher level shapes like trees, see Figure 2. Additionally, these objects were scaled, mirrored and rotated slightly. The shape templates were taken from a dataset that has been published on the internet[4], cf. Figure 2.

[4] `https://www.etsy.com/de/listing/726267122/baum-silhouette-svg-bundle`

Our dataset is too small to train the two neural networks. Therefore, we worked with pre-trained models of DeepFill v2 and GMCNN. In both cases, the pre-training was based on images of the “places2”[5] [29] dataset that had a resolution of 256×256 pixels. In this model as well as in Figures 2 and 4, all pixels with the color white (RGB $(255, 255, 255)$) define a mask. Masks can represent more than one freely shaped object. We also worked with GMCNN pre-trained on “Paris streetview” data [4] but observed artifacts along the boundaries of occluded regions so that we went on only with the “places2” model. Since only part of the images in “places2” show facades, we applied transfer learning with images from the “Ecole Centrale Paris Facades Database”[6], “FaSyn13”[7] [3], and “CMP”[8] [22] datasets. These images were scaled to have the same number of 512 rows. If the number of columns exceeded 512, an image was cut into several frames. If a width was less than 512 columns, the image was expanded by means of mirroring. We also tried transfer-learning with images having 256 columns and rows, but, at least with DeepFill v2, transfer-learning with images having a resolution of 512×512 pixels led to much better results.

The training dataset contained $1,600$ images divided into $1,440$ training images and 160 images for validation. Whereas the three source datasets are widely used to compare the performance of facade related algorithms (e.g., for instance segmentation), their images do not origin from oblique aerial imaging such that, e.g., their resolution is higher and background like the sky is visible.

To train DeepFill v2, we equipped the training and validation images with the automatically generated occlusion masks. GMCNN allows for automatic training with randomly chosen rectangles as occluded areas. For simplicity, we did not change the code but used this training option.

Both neural networks ran on an NVIDIA P6000 GPU. DeepFill v2 was trained with a batch size of eleven due to hardware limitations. As proposed in [27], the adversarial loss and the l_1 reconstruction loss were equally weighted in one test scenario, denoted with (S1). For this scenario, $74,000$ training steps were executed in 18.5 epochs in 90 hours. Since we compare inpainted images with ground truth images in an l_2 norm that is equivalent to the l_1 norm in a finite dimensional space, we did a second test in which we chose a higher weight factor for the l_1 loss by multiplying this loss with 1.1 whereas the adversarial loss was only weighted with 0.9. This scenario is denoted with (S2). To train the scenario, $134,000$ training steps were executed in 33.5 epochs in 160 hours.

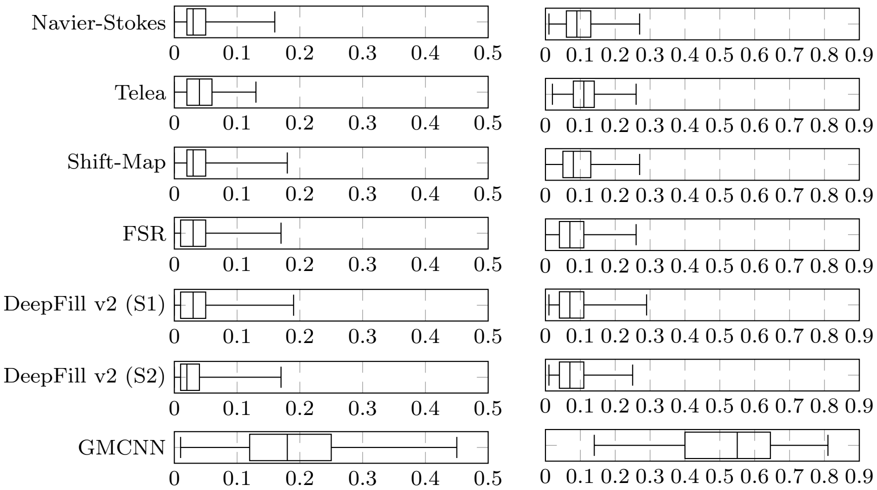
GMCNN was trained with a batch size of 32 and $60,000$ training steps were executed in 60 epochs in 120 hours.

[5] http://places2.csail.mit.edu/download.html

[6] http://vision.mas.ecp.fr/Personnel/teboul/data.php

[7] http://people.ee.ethz.ch/~daid/FacadeSyn/

[8] https://cmp.felk.cvut.cz/~tylecr1/facade/



(a) distances measured with (1)

(b) distances measured with (2)

Fig. 3. Distribution of distances between ground truth and inpainted images of the entire test dataset; DeepFill v2 (S1) relates to equally weighted loss components and DeepFill v2 (S2) shows the result for a higher weighted l_1 loss

5 Results

Figure 4 shows the inpainting results for several facade images and occlusion masks. The Navier-Stokes and Telea algorithms were applied with a circular neighborhood having a radius of 128 pixels. Unfortunately, the openCV xphoto beta version of the FSR algorithm had memory allocation problems and failed to compute results for some images. In contrast to the other algorithms, we therefore tested FSR with reduced image sizes. First, we trained both DeepFill v2 scenarios (S1) and (S2) for 90 hours. Since the results of (S2) were visually better than those of (S1), we extended (S2) training to 160 hours as described before.

We compared the output images with the corresponding ground truth images by simply applying an l_2-norm, for other metrics cf. [16]. Let $G, A \in \{0, 1, \dots, 255\}^{m \times n \times 3}$ be a ground truth image G and an inpainted image A with m rows, n columns and three channels. Let M be the set of coordinates of all masked pixels with $|M|$ elements. Then we measured the distance between A and G via two distance metrics

$$\mathrm{dist}_{\mathrm{all}}(A, G) := \sqrt{\frac{\sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{3} (A_{i,j,k} - G_{i,j,k})^2}{m \cdot n \cdot 3 \cdot 255^2}}, \tag{1}$$

$$\mathrm{dist}_{\mathrm{mask}}(A, G) := \sqrt{\frac{\sum_{(i,j) \in M} \sum_{k=1}^{3} (A_{i,j,k} - G_{i,j,k})^2}{|M| \cdot 3 \cdot 255^2}}. \tag{2}$$

Table 1. Distance between ground truth images and inpainted images, see equation (1); the image numbers refer to Figure 4. Bold and underlined numbers indicate best and second best results

Image	Navier-Stokes	Telea	Shift-Map	FSR	DeepFill v2 (S1)	DeepFill v2 (S2)	GMCNN
1	0.026	0.05	0.024	**0.007**	0.013	$\underline{0.012}$	0.255
2	0.022	0.023	0.033	**0.019**	$\underline{0.02}$	0.022	0.257
3	0.043	0.045	0.077	0.037	$\underline{0.032}$	**0.028**	0.096
4	0.027	0.03	0.028	0.018	$\underline{0.016}$	**0.014**	0.168
5	0.071	0.066	$\underline{0.057}$	**0.05**	0.073	0.061	0.194
6	0.097	0.088	0.134	0.121	$\underline{0.084}$	**0.079**	0.211
7	0.038	0.044	0.051	**0.025**	0.027	$\underline{0.026}$	0.15
8	0.035	0.036	0.034	**0.025**	0.031	$\underline{0.03}$	0.163
9	0.094	0.088	**0.077**	0.097	0.089	$\underline{0.086}$	0.26
10	0.081	0.079	0.083	**0.073**	0.077	**0.073**	0.198
11	0.093	0.087	0.108	**0.075**	0.099	$\underline{0.08}$	0.259
12	0.047	0.044	**0.025**	0.05	0.034	$\underline{0.031}$	0.127
13	$\underline{0.005}$	0.009	$\underline{0.005}$	**0.003**	0.006	0.006	0.262

These distances are normed to be in the interval $[0, 1]$. The box plots in Figure 3 show how the distances are distributed for each algorithm. A high distance value might indicate a bad inpainting result but a pixel-wise comparison might also lead to significant distances although the images appear similar. Instead of using more sophisticated metrics to measure similarity, the quantitative evaluation can be accompanied by a visual qualitative inspection. For example, the Shift-Map algorithm copies rectangular structures that appear consistent even if they do not fit. But a blurred region attracts attention even if it is closer to the ground truth. For the images in Figure 4, distances dist_{all} are listed in Table 1 and distances $\text{dist}_{\text{mask}}$ are shown in Table 2. While $\text{dist}_{\text{mask}}$ only measures how well mask regions can be reconstructed, dist_{all} also takes into account changes outside the mask region, e.g., along its border. The data show that the algorithms focus the changes to the mask regions. Although the distance measures are normalized with respect to the image size, a comparison of the FSR values calculated on downscaled images with those of the other algorithms is somewhat limited. In GMCNN results, inpainted regions tend to show a slightly different color distribution. This leads to large distance values. We did not investigate if better results can be obtained with a different choice of hyperparameters and training data.

Figure 2 illustrates the influence of different mask types. If the hidden regions were not described by completely filled contours but were interrupted by many non-hidden points, most algorithms worked better.

6 Conclusions

DeepFill v2 delivered excellent results, but also the Shift-Map algorithm performed well. This is expected to be true for the FSR algorithm as well, once a stable implementation is available. As shown in [14], the EdgeConnect GAN [16] also is suitable for facade inpainting. The algorithms can be used without

Table 2. Mask-specific distance between ground truth images and inpainted images corresponding to Table 1, see equation (2)

Image	Navier-Stokes	Telea	Shift-Map	FSR	DeepFill v2 (S1)	DeepFill v2 (S2)	GMCNN
1	0.06	0.117	0.051	**0.014**	0.029	0.027	0.589
2	0.059	0.061	0.086	**0.047**	0.054	0.059	0.696
3	0.126	0.133	0.215	0.097	0.094	**0.082**	0.283
4	0.09	0.107	0.089	0.055	0.053	**0.049**	0.574
5	0.168	0.157	0.13	**0.108**	0.174	0.146	0.463
6	0.194	0.175	0.263	0.228	0.168	**0.157**	0.42
7	0.123	0.143	0.155	**0.073**	0.09	0.086	0.492
8	0.115	0.119	0.107	**0.073**	0.102	0.101	0.542
9	0.18	0.17	**0.145**	0.178	0.17	0.166	0.5
10	0.219	0.213	0.222	**0.19**	0.211	0.198	0.537
11	0.175	0.164	0.2	**0.136**	0.185	0.149	0.485
12	0.136	0.128	**0.069**	0.13	0.098	0.089	0.365
13	0.016	0.026	0.013	**0.006**	0.017	0.018	0.795

problem specific adjustments. We tested with low resolution facade textures of a real city model but trained with datasets of higher resolution images. It may be possible to enhance the neural network output further by adding low-resolution images to the training dataset. To achieve better results, the datasets could be additionally improved by adding noise and changing the brightness of the images.

There seems to be no longer a need for highly specialized facade inpainting algorithms as referenced in Section 2, so a direct comparison would be interesting.

Acknowledgements

This work was supported by a generous hardware grant from NVIDIA. The authors thank Udo Hannok from the cadastral office of the city of Krefeld for providing the oblique aerial images. The authors are also grateful to Regina Pohle-Fröhlich for valuable comments.

References

1. Bertalmio, M., Bertozzi, A., Shapiro, G.: Navier-Stokes, fluid dynamics, and image and video inpainting. In: Proc. CVPR 2001 (2001). https://doi.org/10.1109/CVPR.2001.990497
2. Cohen, A., Oswald, M.R., Liu, Y., Pollefeys, M.: Symmetry-aware façade parsing with occlusions. In: Proc. 2017 International Conference on 3D Vision (3DV). pp. 393–401 (2017). https://doi.org/10.1109/3DV.2017.00052
3. Dai, D., Riemenschneider, H., Schmitt, G., Van, L.: Example-based facade texture synthesis. In: Proc. 2013 IEEE International Conference on Computer Vision (ICCV). pp. 1065–1072. IEEE Computer Society, Los Alamitos, CA (2013)
4. Doersch, C., Singh, S., Gupta, A., Sivic, J., Efros, A.: What makes Paris look like Paris? ACM Trans. Graph. **31**(4:101), 1–9 (2012)
5. Genser, N., Seiler, J., Schilling, F., Kaup, A.: Signal and loss geometry aware frequency selective extrapolation for error concealment. In: Proc. 2018 Picture Coding Symposium (PCS). pp. 159–163 (2018)

6. Goebbels, S., Pohle-Fröhlich, R.: Roof reconstruction from airborne laser scanning data based on image processing methods. ISPRS Ann. Photogramm. Remote Sens. and Spatial Inf. Sci. **III-3**, 407–414 (2016)
7. Goebbels, S., Pohle-Fröhlich, R.: Automatic unfolding of CityGML buildings to paper models. Geographies 1 (3) **1**(3), 333–345 (2021)
8. He, K., Sun, J.: Statistics of patch offsets for image completion. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) Proc. 12th European Conference on Computer Vision. Lecture Notes in Computer Science, vol. 7573, pp. 16–29. Springer, Berlin (2012)
9. Hensel, S., Goebbels, S., Kada, M.: LSTM architectures for facade structure completion. In: Proc. GRAPP 2021. pp. 15–24 (2021)
10. Huang, J.B., Kang, S.B., Ahuja, N., Kopf, J.: Image completion using planar structure guidance. ACM Trans. Graph. **33**(4), 1–10 (2014)
11. Kaup, A., Meisinger, K., Aach, T.: Frequency selective signal extrapolation with applications to error concealment in image communication. AEUE – International Journal of Electronics and Communications **59**, 147–156 (2005)
12. Korah, T., Rasmussen, C.: Spatiotemporal inpainting for recovering texture maps of occluded building facades. IEEE Transactions on Image Processing **16**(9), 2262–2271 (2007)
13. Kottler, B., Bulatov, D., Zhang, X.: Context-aware patch-based method for façade inpainting. In: Proc. GRAPP 2020. pp. 210–218 (2020)
14. Kottler, B., List, L., Bulatov, D., Weinmann, M.: 3GAN: A three-GAN-based approach for image inpainting applied to the reconstruction of occluded parts of building walls. In: Proc. VISAPP 2022 (2022)
15. Mehra, S., Dogra, A., Goyal, B., Sharma, A.M., Chandra, R.: From textural inpainting to deep generative models: An extensive survey of image inpainting techniques. Journal of Computer Science **16**(1), 35–49 (2020)
16. Nazeri, K., Ng, E., Joseph, T., Qureshi, F.Z., Ebrahimi, M.: EdgeConnect: Generative image inpainting with adversarial edge learning. arXiv preprint **1901.00212** (2019)
17. Pritch, Y., Kav-Venaki, E., Peleg, S.: Shift-map image editing. In: Proc. 2009 IEEE International Conference on Computer Vision (ICCV). pp. 151–158. IEEE Computer Society, Los Alamitos, CA (2009)
18. Seiler, J., Jonscher, M., Schöberl, M., Kaup, A.: Resampling images to a regular grid from a non-regular subset of pixel positions using frequency selective reconstruction. IEEE Transactions on Image Processing **24**(11), 4540–4555 (2015)
19. Shen, H., Li, X., Cheng, Q., Zeng, C., Yang, G., Li, H., Zhang, L.: Missing information reconstruction of remote sensing data: A technical review. IEEE Geoscience and Remote Sensing Magazine **3**(3), 61–85 (2015)
20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proc. 3rd International Conference on Learning Representations (ICLR) 2015. San Diego, CA, USA (2015)
21. Telea, A.: An image inpainting technique based on the fast marching method. Journal of graphics tools **9**(1), 23–34 (2004)
22. Tyleček, R., Šára, R.: Spatial pattern templates for recognition of objects with regular structure. In: Hein, M., Schiele, B. (eds.) Proc. German Conference on Pattern Recognition (GCPR). Lecture Notes in Computer Science, vol. 8142, pp. 364–374. Springer, Berlin (2013)
23. Wang, Y., Tao, X., Qi, X., Shen, X., Jia, J.: Image inpainting via generative multi-column convolutional neural networks. In: Proc. 32nd International Conference on

Neural Information Processing Systems. p. 329–338. NIPS'18, Curran Associates Inc., Red Hook, NY (2018)

24. Wonka, P., Wimmer, M., Sillion, F., Ribarsky, W.: Instant architecture. ACM Trans. Graph. **22**(3), 669–677 (2003)
25. Yeh, Y.T., Breeden, K., Yang, L., Fisher, M., Hanrahan, P.: Synthesis of tiled patterns using factor graphs. ACM Trans. Graph. **32**(1), 1–13 (2013)
26. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.: Generative image inpainting with contextual attention. In: Proc. CVPR 2018, arXiv preprint 1801.07892 (2018)
27. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.: Free-form image inpainting with gated convolution. In: Proc. CVPR 2019, arXiv preprint 1806.03589 (2019)
28. Zhang, J., Fukuda, T., Yabuki, N.: Automatic object removal with obstructed façades completion using semantic segmentation and generative adversarial inpainting. IEEE Access **9**, 117486–117495 (2021)
29. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: A 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **40**(6), 1452–14649 (2018)

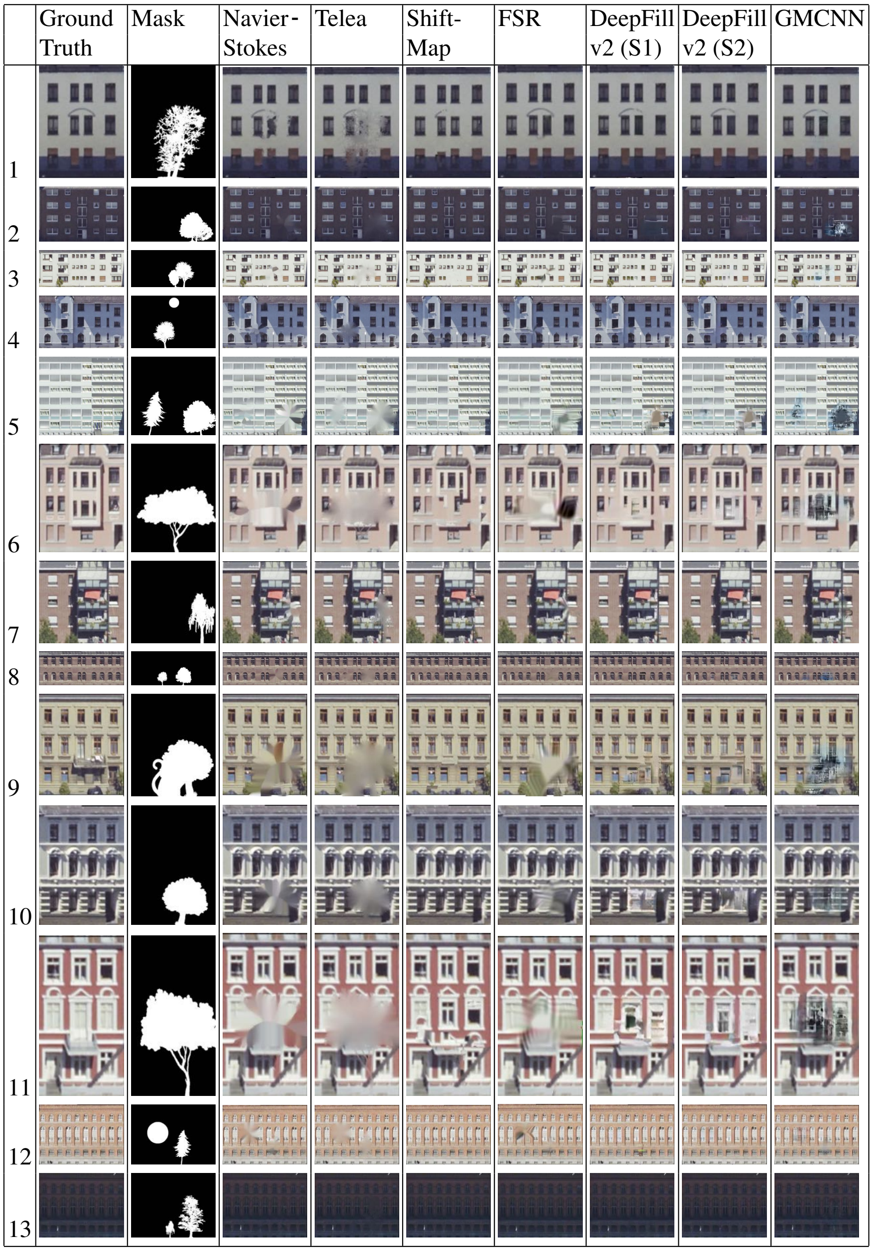


Fig. 4. Comparison of inpainting algorithms; DeepFill v2 was trained with equal weighted loss functions (S1) and with l_1 loss weighted higher than GAN loss (S2)