Context-aware patch-based method for façade inpainting

Benedikt Kottler[1], Dimitri Bulatov[1], Zhang Xingzi[2]
[1]*Department of Scene Analysis, Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB), 76275 Ettlingen, Germany*
[2]*Fraunhofer Singapore, Singapore*
benedikt.kottler@iosb.fraunhofer.de, dimitri.bulatov@iosb.fraunhofer.de, zhang.xingzi@fraunhofer.sg

Keywords: City modeling, inpainting, patch-based inpainting, Textures, Texture image, Facade elements

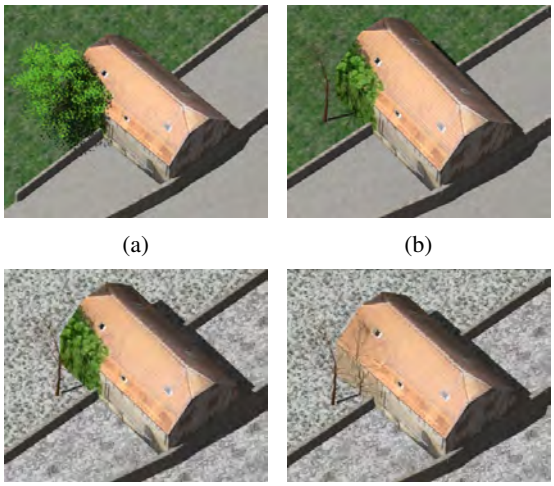
Abstract: Realistic representations of 3D urban scenes is an important aspect of scene understanding and has many applications. Given untextured polyhedral Level-of-Detail 2 (LoD2) models of building and imaging containing façade textures, occlusions caused by foreground objects are an essential disturbing factor of façade textures. We developed a modification of a well-known patch-based inpainting method and used the knowledge about façade details in order to improve the façade inpainting of occlusions. Our modification focuses on suppression of undesired, superfluous repetitions of textures. To achieve this, a coarse inpainting result by a structural-based method is used to influence the choice of the best patch so that homogeneous regions are preferred. The coarse inpainting is calculated using the context knowledge and average color instead of traditionally applied arbitrary structural inpainting. Our modification furthermore introduces a parameter that allows to weight the influence of the coarse inpainting. A parameter study shows that this parameter can be chosen intuitively and does not require any parameter choice method. The cleaned façade textures could be successfully integrated into the accordingly adjusted building models thus upgrading them to LoD3.

1 INTRODUCTION

Realistic representations of 3D urban scenes, in particular, buildings walls from openly available images is an important aspect of scene understanding and has many applications in rapid response missions (Bulatov et al., 2014a; Bulatov et al., 2014b) and virtual tourism (Shalunts et al., 2011), because buildings are becoming easier to identify. For other applications, such as computation of energy balance and, more in general, simulation of buildings in spectral ranges other than visual (Guo et al., 2018), it is important to know the materials of façade parts and not only textural compound. The ability to open, close and breach windows may be useful in applications connected with virtual reality (Bulatov et al., 2014b).

Summarizing, plausible decomposition of façade models into classes e.g. door, window, background using only a minimum of information, namely, texture images from data-bases are meaningful requirements. Such images are available through Internet-based services (Google Maps, Bing Maps, and others) providing public access to geographic information system (GIS) data as well. Hence, there are many related works (Vanegas et al., 2010; Zhang et al., 2019) striving for adding automatically the 3D geometry of the observed buildings to the building models. In particular, in the simple and robust approach of (Zhang et al., 2019), dedicated to extending textured LoD2 models to LoD3 models, images were rectified, the façade elements like doors and windows detected, and the 3D model updated. However, in the *Berlin CityGML* dataset employed for this work, the quality of the texture images was rather poor. Common problems in low-quality texture images, such as lens distortions, overexposure/underexposure, illumination inconsistencies, shadows/occlusions, reflection/refraction, etc., can cause a significant reduction in the rendering quality of a 3D city scene. Furthermore, since the texture images of building facades were the only input of the procedure of (Zhang et al., 2019), proper occlusion analysis could not be carried out. This had a disadvantage that the façade images were often contaminated, among others and in particular, by trees standing in front of the buildings. Especially when it comes to simulation of winter scenes, unpleasant and unrealistic green stains at building walls as in Figure 1 are clearly visible.

In order to prevent this from happening, we wish to extend the approach of (Zhang et al., 2019) into the direction of cleaning building walls from textures



(a) (b) (c) (d)

Figure 1: In summer captured textures need adaptation for winter environments. (a) Summer scene with modeled tree, (b) Summer scene without modeled tree, (c) Winter scene with distracting texture, (d) Winter scene with cleaned texture

of trees using inpainting methods. The texture-based inpainting is identified as a state-of-the-art tool to fill large holes in the images. Our main contribution is that the context, retrieved at the stage of façade detection of (Zhang et al., 2019), is extensively taken into account. Note that we assume here that the masks for trees and possibly other foreground objects are given because otherwise they can be extracted using deep learning techniques for semantic segmentation, such as (Chen et al., 2018).

2 Previous Approaches

This section will consist of two parts, dedicated respectively to particularly representative approaches for creation of LoD3 models, or, more specifically, detection of façade elements, and those for image inpainting.

2.1 Creation of LoD3 models

We start our selection of representative methods on façade analysis by the method of (Xiao et al., 2008), who use actual sensor data given as image sequences from ground-based vehicles. A façade is recursively decomposed into a set of rectangular patches based on the horizontal and vertical lines detected in the texture image. The next step is constituted by a bottom-up merging of patches, whereby the authors make use of the architectural bilateral symmetry and repetitive patterns automatically detected. Finally, a vertical plane with independent depth offset is assigned to each patch, however, without assigning a semantic label. The authors of (Loch-Dehbi and Plümer, 2015) rely on a statistical approach. They automatically generate a small number of most likely hypotheses and provide the corresponding probabilities. The model parameters can be retrieved from Gaussian mixture distributions, whereby constraints between these parameters are simultaneously taken into account. Probably, the main risk of this method could ending up in the combinatoric chaos. An interesting approach is presented by (Guo et al., 2018). The façade is subdivided into facets. To determine whether a facet contains a window, R-CNNs (Ren et al., 2015) are employed. Even though this approach is not evaluated, the application field is interesting: Representation of the digital twin in thermal infrared spectrum. Yet another approach (Wenzel and Förstner, 2016) contains an optimization procedure incorporated into monte-carlo-like simulation with multiple births and deaths of objects (doors, windows, and balconies). The regularization prior is given by symmetry constraints of façade objects and the data term is constituted by descriptors allowing detecting corners and other salient structures in images. Despite impressive results, monto-carlo simulation and optimization by a procedure reminding simulated annealing is costly. To prove how the proposed context-aware inpainting works in challenging situations, we decided to assess a method of (Zhang et al., 2019), which provides good results without too many complex computations. In Section 3, we will provide a short overview of this work.

2.2 Image inpainting techniques

Concerning inpainting techniques, which represents the focus of this paper, most of them can be denoted as either structure-based or texture-based. To the first subgroup, we assign the methods based on kriging (Rossi et al., 1994), marching method of (Telea, 2004), or on finding a stationary solution of a partial differential equation. The variational component is added to enforce a model assumption, such as piecewise smoothness. Thus, the relevant methods (Bertalmio et al., 2001; Schönlieb and Bertozzi, 2011) are usually employed to tackle simultaneous the denoising and inpainting task. The structure-based methods have been intensively studied during many years because of their inter-disciplinary applicability. However, it has been established that they neither perform well in case of richly textured background nor for too large holes, since this provokes

as a consequence unnatural blurring of the inpainting domain. Therefore, starting at epochal publications of (Efros and Leung, 1999; Criminisi et al., 2004), texture-based methods have become increasingly popular in the recent years. The basic idea is to search for every pixel to be inpainted a window, which partly appeared somewhere else and there context is filled. Countless modifications have been implemented since then, including manipulations of priority terms for pixels to be inpainted, e.g. (Le Meur et al., 2011), convex combinations of patches in order to suppress noise and artifacts between pasted patches, and narrowing search spaces (Buyssens et al., 2015). The crucial leverage, is hidden in this ''somewhere else". In remote sensing, the didactically excellent paper of (Shen et al., 2015), differentiate between spatial, spectral, and temporal techniques. They search for the missing information in other parts of the same image, in another channel, or, respectively, even in some completely different images. This last group obviously opens the door to CNN-based method which now are being increasingly employed for inpainting tasks as well. From the pioneering work of (Yeh et al., 2017), who were able to complete images with only a very sparse number of pixels covered using deep learning, only a few years have passed until the NVidia solution (Liu et al., 2018) became available online.

Typical for most solutions based on deep learning is that the context information is sacrificed in favor of millions of training examples. The authors, however, decided to make use of context and to process not only the raw images from (Zhang et al., 2019), but also the classification results of façade elements. In case of structure-based inpainting, (Kottler et al., 2016) demonstrated that even a simple method allows to preserve sharp edges between classes, once context information is taken into account. This paper will describe a suitable modification of the approach of (Criminisi et al., 2004).

3 PRELIMINARIES

In this section, we will first briefly describe the procedure of (Zhang et al., 2019), in which applying inpainting techniques are required. In the second subsection, we introduce the original method of Criminisi et al. In the third subsection, a CNN-based inpainting as a state-of-the-art tool will be outlined for comparison.

3.1 Texture analysis and creating LoD3 models of Zhang et al.

In what follows, we describe the approach of (Zhang et al., 2019) which starts at LoD2 CityGML and texture images and aims at obtaining 3D models of the façade, which are enriched with semantics about important façade elements like doors and windows. This is a data-driven approach which allows to generate plausible LoD3 models from textured LoD2 models. This method starts by the rectification of a texture image to make the facade elements axis-parallel. Then, automatic detection of doors and windows in the texture images is based on the Mask R-CNN method within ResNet. There are two successive phases in Mask R-CNN. A Region Proposal Network proposes candidate object bounding boxes and an Object Detection Network and a Mask Segmentation Network predict the class and box offset. Thus, the second phase has the output of the first one phase as input and the combined loss function takes both losses into account. After the detection, rule-based clustering is applied to align the detected façade elements and regularize their sizes. The features for clustering are the coordinates of object centroids. The sizes of the façade elements are taken into account as well. For example, if the size of a façade element differs from other façade elements, it is marked as outlier and would not be regularized. After interactively selecting matched models from a 3D facade element model database, deforming and stitching them with the input LoD2 model takes place, as Figure 2a shows.

In this way, building models can be extended from LoD2 to LoD3 based on its texture images. However, the texture images of the LoD2 CityGML models are usually aerial images taken from top or side views. As shown in Figures 2a and 2b, their quality is sometimes very poor. Common problems in low-quality texture images include distortions, overexposure/underexposure, illumination inconsistencies, shadows/occlusions, reflection/refraction, redundancy and etc. These problems can cause a significant reduction in the rendering quality of a 3D city scene. The present work aims at overpainting the foreground objects, like trees, from the images before adding them to the model.

3.2 Texture-based inpainting of Criminisi et al.

In this section, we briefly describe the algorithm of (Criminisi et al., 2004) and introduce the relevant modifications later.

Given is an image with image region I, region to

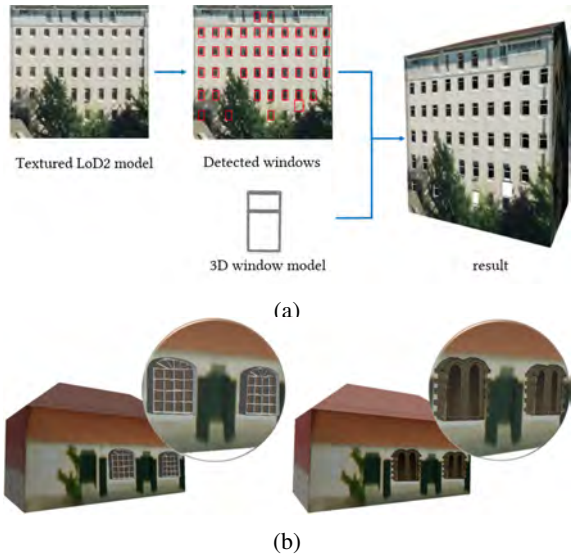


Figure 2: (a) Pipeline of the approach of (Zhang et al., 2019), (b)s examples of extending LoD2 model to LoD3. As we can see, the texture images in both (a) and (b) are negatively affected by foreground objects (trees).

be filled $D \subset I$ and a source region $\mathcal{S} \subset I - D$. Straight forward, S may be defined as the entire image minus the target region ($\mathcal{S} = I - D$), or it may be manually specified by the user. The inpainting evolves inward as the algorithm progresses, and so we also denote to ∂D as the fill front. Moreover, as with all exemplar-based texture synthesis (Efros and Leung, 1999), the size of the template window $\mathcal{S}$ must be specified. The authors of (Criminisi et al., 2004) suggest a 9×9 window as default.

Each pixel maintains a confidence value $C(p)$, which reflects reliability of the pixel information, and which is frozen once a pixel has been filled. During the algorithm, patches along the fill front are also given a temporary priority value, which determines the order in which they are filled.

The algorithm fills the pixels around the filling front one after the other in order of prioritization by copying image content from the best patch. The patch of thé current pixel of the filling front is compared with all patches in the source area S. Using a cost function evaluated on the pixels outside the inpainting area, the patch yielding the minimum distance is then copied to the target area. The core of the algorithm is an isophote-driven prioritization process that combines structure and texture inpainting.

The method of (Criminisi et al., 2004) has the restriction that, applied on images with complicated contents and repetitive structures may be copied into the inpainting area. Our proposed method is supposed to smooth the result and how to use context knowledge to restore the known objects.

3.3 Inpainting using partial convolution neuronal network of Liu et al.

Deep learning methods are currently very popular and are successfully employed in many areas of computer vision. Therefore, we use an inpainting method (Liu et al., 2018) as a comparison to our proposed method. This deep neural network has a U-Net-like structure with deconvolutions based on nearest-neighbor principle. The inpainting problematic is reflected by replacing all convolution layers by the so-called partial convolution layers, that is, only pixels within the inpainting mask are affected by convolutions. We used the implementation available on Github [1] which included a pre-trained model with images of size 256×256, trained for about 1 day on a small part of ImageNet. For comparison, the original paper has image sizes 512×512 and ten days training for ImageNet and Places2 and three days for CelebA-HQ. Input images are resized on the shorter side to 256 pixels with maintaining the original aspect ratio. The selection of a square with 256×256 took place, whereby cropping is performed the way the door is preserved in order not to lose important façade details.

4 METHOD

4.1 Inpainting with classes

The two critical parts of the algorithm (Criminisi et al., 2004) are determining the priorities, which should be made as smart as possible, and the finding of the best copy patch, which should be done as efficiently as possible.

The aforementioned prioritization prefers pixels which (i) are a continuation of edges and (ii) are surrounded by many pixels containing information. We search for the maximum of the priority term which is given as a product

$$Pr(p) = Co(p)Da(p), \tag{1}$$

where

$$Co(p) = \frac{\sum_{q \in \mathcal{N}_p \cap (I-D)} Co(q)}{|\mathcal{N}_p|}, \qquad Da(p) = \frac{\nabla I_p^{\perp} \cdot n_p}{\beta}$$

and $|\mathcal{N}_p|$ denotes the number of neighboring pixels. Initially, all pixels in the inpainting area bear no information content, so $Co(p) = 0$ for all $p \in D$ and all

[1] https://github.com/ja3067/pytorch-inpainting-partial-conv

pixels outside have the value 1, so $Co(p) = 1$ for all $p \in (I - D)$. In the following, pixels are preferred which have as many already filled neighbors as possible, whereas pixels, which were filled earlier, are preferred. The data term $Da(p)$ describes the strength of isophotes hitting the front of D. As filling proceeds, this term boosts the priority of edges entering the inpainting domain and gives the algorithm its structural inpainting component.

After the prioritization step, the pixel q with the best copy patch is determined for the patch of the pixel p with the highest priority. The aim is to find the patch which is closest to the given patch. The distances between the target patch and the patches of the pixels in the source region are calculated and a minimum argument is determined, so

$$q^* = \arg\min_{q \in \mathcal{S}} d(p,q),$$

where the distance measure is denoted with $d : I \times I \to \mathbb{R}$.

It is defined as follows:

$$d(q,p) = \sum_{x_q \in \mathcal{N}_q} \{[\psi(x_q) - I(y_p(x_q)] \cdot M(x_q)\}^2$$

where in the general case

$$\psi(x) = \begin{cases} I(x) & \text{if } x \notin D \\ \tilde{I}(x) & \text{if } x \in D \end{cases} \quad (2)$$

or in the case with classes

$$\psi(x) = \begin{cases} I(x) & \text{if } x \notin D \\ \text{Col}(i) & \text{if } x \in \{y \in D | C(y) = i\} \end{cases}, \quad (3)$$

where $\text{Col}(i)$ denotes the average color of class i, and

$$M(x) = \begin{cases} 1 & \text{if } x \notin D \\ \alpha & \text{else} \end{cases} \quad (4)$$

denotes the function ψ corresponds to the source image I outside of the inpainting domain. Inside this domain, it corresponds to an approximation of the image content $\tilde{I}$. To preserve structural changes of color information, $\tilde{I}$ can be a structural inpainting result. Since elements of different classes are present, we decided to break the procedure down to the level of classes. Thus, an approximation is calculated with the classification by setting the color value of a pixel to the average color of the corresponding class (see section 5.2). This approximation is based on additional information, for example the result of structure inpainting, or on a classification result. The parameter $\alpha \in [0,1]$ in M describes the weighting of the approximation introduced in $\tilde{I}$. It can set intuitively as shown in section 5.1.

The algorithm can be seen as a function $I_{Inp} = Cr(I, D, \mathcal{S}[, C])$, where the classification result is optional and only used to calculate the approximation.

4.2 Façade reconstruction

The first input in this section is a rectified façade texture image I in which the foreground objectes are detected. After morphological dilation we obtain the inpainting area D. The second, optional input is the classification result C of the façade images described in Section 3.1. One can obtain a mask for a given class i by

$$C_i(p) = \begin{cases} 1 & \text{if } C(p) = i \notin D \\ 0 & \text{otherwise} \end{cases}$$

For the concrete problem, pixels belonging to façade background and foreground object should be inpainted using only regions of visible façade background, and pixels belonging to façade details should be inpainted using only corresponding pixels of the according class. We denote this set of "valid pixels" by the source region $\mathcal{S} = D^c \cap C_{\text{f. b.}}$ and will apply our method to the problem.

For the façade images, the algorithm was executed twice. The first time, it is used to reconstruct the façade background while the second time it is used to add the façade details to the first result. This leads to a two-step process for façade reconstruction.

Façade background: For the façade background, the inpainting area is restricted to the area belonging to the façade background class. Accordingly, the source area is restricted to the visible area, which belongs to the class façade background. This results in the rule:

$$I_{InpBack} = Cr(I, D \cap C = \text{f. b.}, \mathcal{S} = D^c \cap C_{\text{f. b.}}, C)$$

Façade details: After that, the façade details can be added. The painting area is restricted to the area belonging to the façade background class. Accordingly, the source area is also restricted to the visible area that belongs to the class façade details. This results in the analogical rule:

$$I_{InpFaçade} = Cr(I, D \cap C = \text{f. d.}, \mathcal{S} = D^c \cap C_{\text{f. b.}}, C).$$

Summarizing, in the first step, the facade background was reconstructed. In the second step we add the façade details by repeating the first step, but using only the locations of covered windows as the inpainting area and the areas where doors and windows as source areas lied.

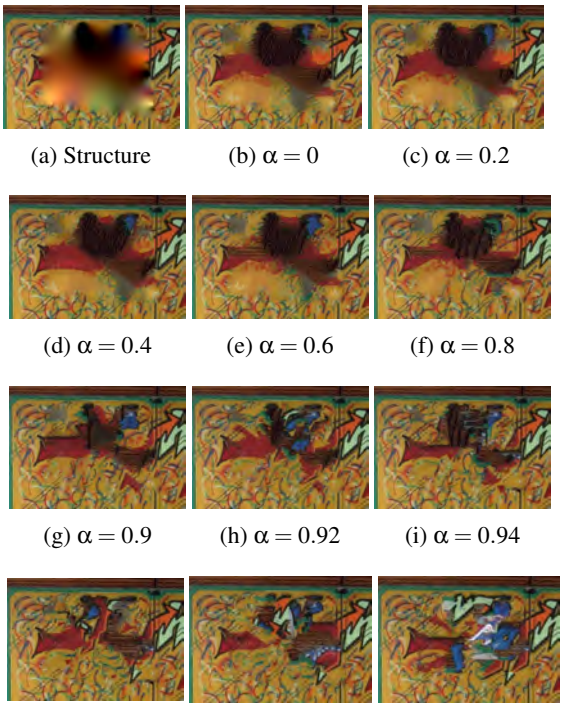
5 RESULTS AND DISCUSSION

In this section, we will first make a parameter study for α in equation (4) and will show that its consid-

eration allows preventing unwanted repetition of patterns while its choice is quite straightforward and intuitive. Then, we present the results of the modified method of (Criminisi et al., 2004) applied on a data set already analyzed by (Zhang et al., 2019). We considered rectified façade texture images, which were part of a LoD3 city model of Berlin and corresponding classification results for windows and doors. For those images, where a foreground object – a tree, for example – was occluded the façade. We selected a binary mask accordingly. Otherwise, a polygon of interest was selected that may best show the differences in performance of all algorithms. Following, a detailed comparison of the results of the proposed method with those produced using the deep learning technique will take place. The section is concluded by a description on a LoD3 model creation.

5.1 Choice of α

In this subsection, we make a visual evaluation of the influence of the parameter α in equation (4) on the inpainting result. For the complex wall texture in the example image, we created a coarse approximation of the structure using Navier-Stokes inpainting, shown in Fig 3a. Then, we run our method and varied α between 0 and 1. In Figure 3, the result for different α is shown. It becomes clear that for α values close but not equal to 0, the dominant colors in the inpaining domain are similar to those around it and that the undesired repetitions of patterns are much more rarely present. The advantage of our method is that texture content is searched for in the regions with similar colors. The blurring is typical for structural inpainting and thus, small values of α, but it disappears gradually with its growing value.



(a) Structure (b) $\alpha = 0$ (c) $\alpha = 0.2$

(d) $\alpha = 0.4$ (e) $\alpha = 0.6$ (f) $\alpha = 0.8$

(g) $\alpha = 0.9$ (h) $\alpha = 0.92$ (i) $\alpha = 0.94$

(j) $\alpha = 0.96$ (k) $\alpha = 0.98$ (l) $\alpha = 1$

Figure 3: Influence of parameter α. The choice $\alpha = 1$ corresponds to the original method

5.2 Façade reconstruction

The results of the façade inpainting are shown by ten characteristic examples of façade images and classification results due to (Zhang et al., 2019) in Figure 4. The classes for doors and windows are represented by blue and red colors, respectively. The patch size for the proposed texture-based method was 21×21 pixels and $\alpha = 0.98$ for all images. As inpainting area, we used the detected foreground object enlarged by applying a morphological dilation.

We believe that in the majority of cases the inpainted – either using the proposed or CNN-based method – images (in the two right columns) have a more realistic appearance than the corresponding original images on the left. We can see that the structural inpainting can preserve the dominant colors of the façade and its elements. In the majority of cases, the course of shadow is dropped (a desirable effect) and in the few cases the shadows are *preserved*, it happens because of progressive propagation of colors during inpainting of texture. The backlashes of our method are those single dark spots which stem from the inaccuracies of the classification result: The part of the façade close to a window is sometimes propagated by pixels stemming from a window, but spuriously classified as wall.

5.3 Comparison with CNN-based approach

In this section, we compare the result of our algorithm with the state-of the-art CNN based method presented in section 3.3. It is important to note that the net not only fills the hole but affects the whole image, which happens because before application, the (ten) test images must be radiometrically and geometrically corrected. By comparing both right-most columns of Figure 4 we see, that our method better reflects the facade elements. The CNN-based method often produced wrongly reconstructed shapes of doors and windows. The undesired effect of course of shadows, which are clearly not present in original wall texture, is more visible here. Also the blurring artifacts are ev-

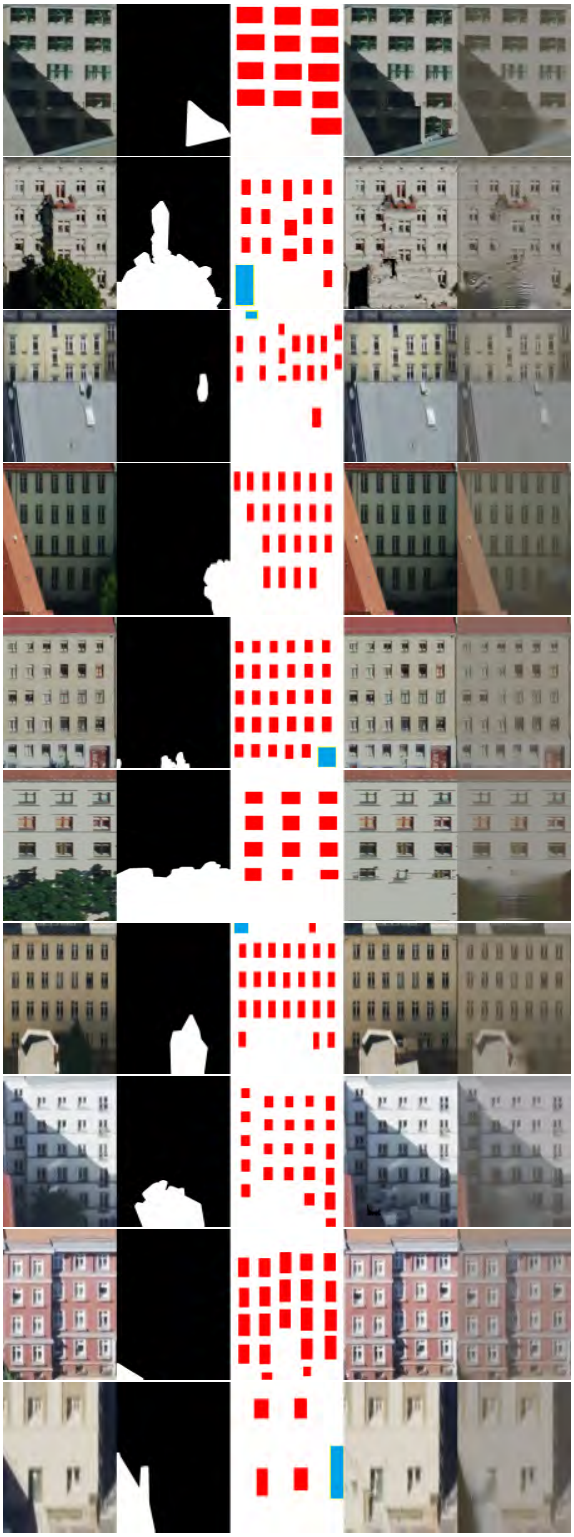


Figure 4: Overview of the data set and results. From left to right: original images, occlusion masks, inpainting using our method, and result of CNN-based inpainting method. Please note that some images were rescaled in order to better fit the page (among other image 5, shown in page 7).

Figure 5: Detail view of image 5. From left to right: original image, result of modified Criminisi et al., result of CNN based method.

ident. In Figure 5, we show more detailed comparison of both methods by exemplifying the fifth image. We feel confirmed in our observations about blurring and shadow course. While the proposed method nicely reconstructed the bottom row of windows, in the CNN-based result, a door seems to have been hallucinated despite no blue box was detected in the fifth row of Figure 4. But apparently, it was enough that a frontal door was present in many training images. Moreover, the grid structure of the door is caused by the deconvolution based on nearest neighbor. In the middle image, one could argue that the homogeneous texture of the bottom part of the façade combined with some artifacts resulting from the inaccurate classification may appear fake, but we believe that the interactive user could better interpret this result than that on the right.

5.4 Textured 3d model

Figure 6 shows the result included in the LoD3 model output in the pipeline of (Zhang et al., 2019). The façade models in the database may contain concavities (window inserts) and convexities (balconies). The presented inpainting is a suitable method to enhance the result and to improve the visual impression. In contrast to the CNN-based method, it is also suitable for closing very large occlusions, e.g. a tree that covers about one third of the wall. A further advantage in connection with the training of rapid response missions is that only identified façade details are added. Within virtual training and rehearsal of such missions, the participants are supposed to find doors and windows only on the spot where they actually are.

(a) (b)

Figure 6: Example of an inpainted LoD3 model: (a) inpainted model, (b) original model (down)

6 CONCLUSIONS

In quick response and virtual reality applications, it is often important for the user to be able to orient him or herself in the unknown terrain. In many applications, not only a visually appealing model is important, but also the trustworthiness of observed information. We have shown that considering classification results improves the inpainting result. In addition to the original procedure of (Criminisi et al., 2004), we use an approximated inpainting result that includes classes straightforward; in the second step, we control the result by weighting them. Cleaned textures are particularly useful to adjust the scenario for different seasons. An example is shown in Figure 1, a model was adapted for a winter scenario, where the uncleaned textures are particularly disturbing. Another question is whether inpainted texture images help to improve the detection results and clustering results in turn. We compared the detection results of two pairs of original and inpainted texture images. The comparison shows that inpainting does not necessarily improve the detection or clustering results. The paper shows that patch-based methods are in no way inferior to a CNN-based method and thereby, we believe to have proved the methodological principle. Additionally, the method still have potential to exploit. For example, we have concentrated on RGB-color space and have performed inpainting channel-by-channel. However, there are sources in the literature, such as (Cao et al., 2011) that pointed out that there are color spaces better suitable for inpainting tasks. It is also clear that if the instances of the same class have different colors (for example, green, blue, and red window), then color averaging would produce colors not present so far. Thus color averaging can be replaced by identification of accumulation points. With respect to CNN-based generation, it would make sense to use the results obtained by our method as training data in order to improve the results of Sec 5.3. Other CNN-based workflows, including generative adversarial networks, seem to be a promising tool as well. Finally, the mask for occluding object was created manually. This is clearly not affordable in applications and therefore, a unified concept incorporating this step into context-aware segmentation is highly desirable.

ACKNOWLEDGMENTS

Many thanks to the student assistant Ludwig List from Fraunhofer IOSB for providing the NVIDIA neural network inpainting result.

REFERENCES

Bertalmio, M., Bertozzi, A. L., and Sapiro, G. (2001). Navier-stokes, fluid dynamics, and image and video inpainting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001*, volume 1, pages I–355.

Bulatov, D., Häufel, G., Meidow, J., Pohl, M., Solbrig, P., and Wernerus, P. (2014a). Context-based automatic reconstruction and texturing of 3D urban terrain for quick-response tasks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93:157–170.

Bulatov, D., Häufel, G., Solbrig, P., and Wernerus, P. (2014b). Automatic representation of urban terrain models for simulations on the example of VBS2. In *SPIE Security+ Defence*, pages 925012–925012. International Society for Optics and Photonics.

Buyssens, P., Daisy, M., Tschumperlé, D., and Lézoray, O. (2015). Exemplar-based inpainting: Technical review and new heuristics for better geometric reconstructions. *IEEE Transactions on image processing*, 24(6):1809–1824.

Cao, F., Gousseau, Y., Masnou, S., and Pérez, P. (2011). Geometrically guided exemplar-based inpainting. *SIAM Journal on Imaging Sciences*, 4(4):1143–1179.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818.

Criminisi, A., Pérez, P., and Toyama, K. (2004). Region filling and object removal by exemplar-

based image inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212.

Efros, A. A. and Leung, T. K. (1999). Texture synthesis by non-parametric sampling. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, volume 2, pages 1033–1038.

Guo, S., Xiong, X., Liu, Z., Bai, X., and Zhou, F. (2018). Infrared simulation of large-scale urban scene through LOD. *Optics express*, 26(18):23980–24002.

Kottler, B., Bulatov, D., and Schilling, H. (2016). Improving semantic orthophotos by a fast method based on harmonic inpainting. In *9th IAPR Workshop on Pattern Recogniton in Remote Sensing (PRRS), 2016*, pages 1–5.

Le Meur, O., Gautier, J., and Guillemot, C. (2011). Examplar-based inpainting based on local geometry. In *2011 18th IEEE International Conference on Image Processing*, pages 3401–3404.

Liu, G., Reda, F. A., Shih, K. J., Wang, T.-C., Tao, A., and Catanzaro, B. (2018). Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100.

Loch-Dehbi, S. and Plümer, L. (2015). Predicting building façade structures with multilinear Gaussian graphical models based on few observations. *Computers, Environment and Urban Systems*, 54:68–81.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Rossi, R. E., Dungan, J. L., and Beck, L. R. (1994). Kriging in the shadows: geostatistical interpolation for remote sensing. *Remote Sensing of Environment*, 49(1):32–40.

Schönlieb, C.-B. and Bertozzi, A. (2011). Unconditionally stable schemes for higher order inpainting. *Commun. Math. Sci*, 9(2):413–457.

Shalunts, G., Haxhimusa, Y., and Sablatnig, R. (2011). Architectural style classification of building façade windows. In *International Symposium on Visual Computing*, pages 280–289.

Shen, H., Li, X., Cheng, Q., Zeng, C., Yang, G., Li, H., and Zhang, L. (2015). Missing information reconstruction of remote sensing data: A technical review. *IEEE Geoscience and Remote Sensing Magazine*, 3(3):61–85.

Telea, A. (2004). An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1):23–34.

Vanegas, C. A., Aliaga, D. G., and Beneš, B. (2010). Building reconstruction using manhattan-world grammars. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 358–365.

Wenzel, S. and Förstner, W. (2016). Façade interpretation using a marked point process. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:363–370.

Xiao, J., Fang, T., Tan, P., Zhao, P., Ofek, E., and Quan, L. (2008). Image-based façade modeling. *ACM Transactions on Graphics (TOG)*, 27(5):161.

Yeh, R. A., Chen, C., Yian Lim, T., Schwing, A. G., Hasegawa-Johnson, M., and Do, M. N. (2017). Semantic image inpainting with deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5485–5493.

Zhang, X., Lippoldt, F., Chen, K., Johan, H., and Erdt, M. (2019). A data-driven approach for adding façade details to textured LoD2 CityGML models. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP*, pages 294–301.