FacaDiffy: Inpainting Unseen Facade Parts Using Diffusion Models Supplementary Material

Thomas Fröch[*,1], Olaf Wysocki[2], Yan Xia[*,3,5] , Junyu Xie[4], Benedikt Schwab[1], Daniel Cremers[3,5], Thomas H. Kolbe[1]

[1]Chair of Geoinformatics, TUM School of Engineering and Design, Technical University of Munich (TUM), Munich, Germany - (thomas.froech, benedikt.schwab, thomas.kolbe)@tum.de
[2] Photogrammetry and Remote Sensing, TUM School of Engineering and Design, Technical University of Munich (TUM), Munich, Germany - olaf.wysocki@tum.de
[3]Computer Vision Group, TUM School of Computation, Information and Technology, Technical University of Munich (TUM), Munich, Germany - (yan.xia, cremers)@tum.de
[4]Visual Geometry Group, Department of Engineering Science, University of Oxford. Oxford, UK - jyx@robots.ox.ac.uk
[5]Munich Center for Machine Learning, Munich, Germany

- Section 1 gives an overview of all types of conflict maps that are applied for various purposes in this work.
- Section 2 provides further information on hyperparameter settings and the computational resources we leveraged in our experiments.
- Section 3 elaborates on the choice of the text prompt, illustrating its strong influence on the inpainting outcomes.
- Section 4 provides an explanation and additional examples of counterintuitive IoU evaluation results.
- Section 5 provides further visual illustrations to showcase additional inpainting results.

1. Conflict Map Datasets

In this section, we provide more details about the conflict map datasets introduced in the main text, elaborating on their sources and usages in our work, with example visualizations provided.

1.1 Real Conflict Maps

Figure 1 presents a range of real conflict maps derived from existing Level of Detail (LoD)2 building models and corresponding mobile laser scanning point clouds. These maps showcase diverse incomplete instances owing to occlusions by cars, pedestrians, vegetation, *etc.*. Also depicted are areas affected by the presence of extruded facade objects in Figure 1d and 1e.

* Corresponding Author

1.2 Synthetically Generated Conflict Maps

As detailed in Section 3.2 in the main text, we design a scalable pipeline to synthetically generate conflict maps, and leverage them to personalize the pre-trained Stable Diffusion inpainting model. These conflict maps originate from LoD3 building models randomly generated using the Random3Dcity application (Biljecki et al., 2016). Figure 2 displays several examples of synthetically generated conflict maps.

1.3 Conflict Maps Derived from Annotated Images

We deploy conflict maps derived from the CMP database of annotated images (Radim and Radim, 2013) to personalize a pre-trained Stable Diffusion model as part of the ablation studies and to evaluate the inpainting results. Corresponding visualizations can be found in Figure 3.

1.4 Ground-Truth Conflict Maps from LoD3 Models

We also consider ground-truth conflict maps obtained from existing LoD3 models. However, due to the scarcity of LoD3 models, a limited number of conflict maps are obtained, which are then adopted for evaluation purposes only. Figure 4 illustrates a collection of exemplary ground-truth conflict maps derived from LoD3 models.

Figure 5 demonstrates the schematic workflow for generating ground-truth conflict maps from LoD3 models, with key procedures elaborated below.

Principal components and rotation parameters. We ensure proper alignment of each facade with the co-

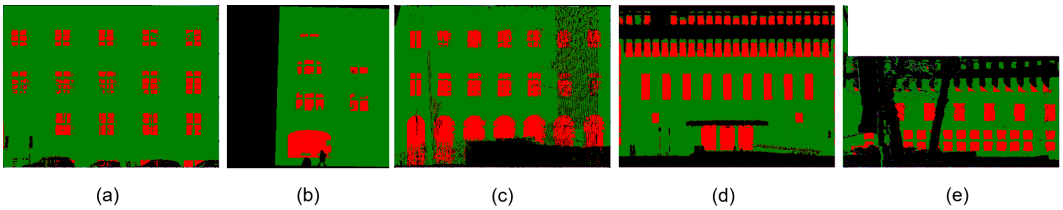

Figure 1. A collection of exemplary conflict maps featuring occlusions by cars (in a), pedestrians (in b), vegetation (in e), missing regions due to lack of coverage (in b), and extruded facade objects (in d and e). Green: Confirming, Red: Conflicting, Black: Unknown / Occluded

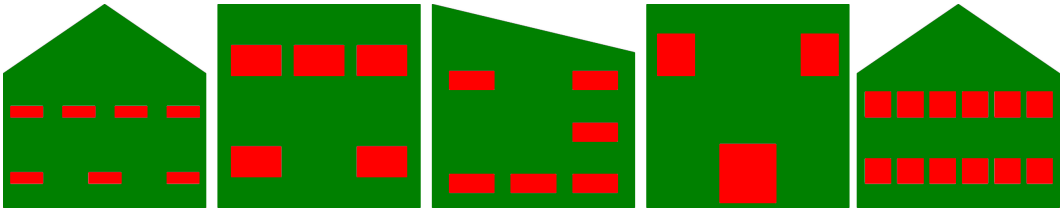

Figure 2. A collection of exemplary synthetic conflict maps derived from semantic LoD3 building models that we randomly generated with Random3Dcity.

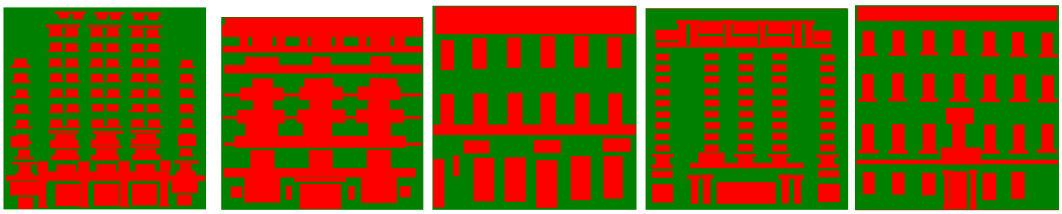

Figure 3. A collection of exemplary conflict maps derived from the CMP database of annotated images.

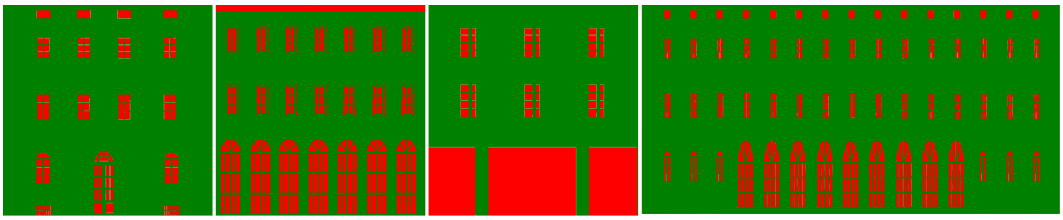

Figure 4. A collection of exemplary ground-truth conflict maps derived from LoD3 models.

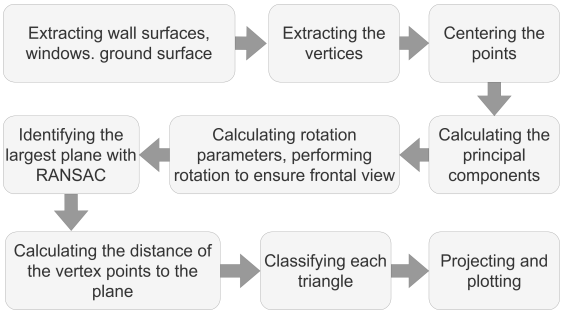

Figure 5. Schematic workflow for generating ground-truth conflict maps from LoD3 semantic building models.

ordinate axes when viewed from the front by evaluating the principal components of the vertex point distribution in centered LoD3 facades. The required rotation angle is then calculated to align the facade with the x-axis of the reference system.

Plane identification with RANSAC. Assuming the majority of vertex points in the LoD3 facade lie in a single plane, we employ the RANSAC algorithm to identify the plane with the highest number of inliers. A threshold of 0.005m and a minimum of 10 inliers are set, with a maximum of 1000 iterations. The implementation in pyRANSAC-3D (Mariga, 2022) is utilized for this process.

Conflict determination. We consider all triangles that lie in the identified plane as confirming. All triangles that deviate from this plane are considered to be conflicting. Due to the voyeur effect (Tuttas and Stilla, 2013), windows are classified as conflicting.

2. Implementation Details

2.1 Conflict Map Computation

In the implementation of our deterministic approach for conflict map computation, we apply the ray-casting functionality implemented in open3d (Zhou et al., 2018). We utilize a dedicated functionality implemented in this library to perform the multi-iteration midpoint triangle subdivision on the triangles the facades comprise.

2.2 Dreambooth Hyperparameters

Since Dreambooth has proven to be sensitive towards the setting of hyperparameters we ensure the comparability of our experiments by consistently deploying the same set of hyperparameter settings thoroughly. We followed the settings proposed by (Patil et al., 2022).

Hyperparameter	Value
Image size	$512 \cdot 512$
Training batch size	1
Number of gradient accumulation steps	1
Learning rate	5×10^{-6}
Number of warmup steps	0
Maximal number of training steps	400
Unique identifier string	“sks”

Table 1. Hyperparameter settings for the personalizations with Dreambooth.

2.3 Computational Resources

We leverage an NVIDIA RTX 8000 GPU with 48 GB VRAM for our personalization experiments with Dreambooth. The personalization process using Dreambooth only takes up to approximately a quarter of an hour.

3. Choice of the Text Prompt

3.1 Static Text Prompt for Experiments and Ablation Studies

To identify a suitable text prompt that is consistently applied for conflict map inpainting, we investigate a variety of text prompts, involving high-level (*e.g., “Window”*) and low-level (*e.g., “Rectangle”*) descriptions and assessing their corresponding effects on the inpainting outcomes.

In these experiments, we observed a strong dependency of the inpainting results on the chosen text prompt. This dependency is vividly illustrated in Figure 6. Qualitatively good outcomes were attained when employing a text prompt that precisely describes the content intended to fill the missing region.

3.2 Automatic Text Prompt-Generation: A Trial

As an alternative approach, our goal is to generate a text prompt automatically based on the following low-level properties of the input images: (i) Histogram; (ii) Symmetry; (iii) Fragmentation.

However, the enhancements achieved through this method are constrained.

Histogram analysis. We identify the background color of the image by analyzing the binary histogram of the input image. The predominant color, characterized by a higher number of occurrences, is considered the background color.

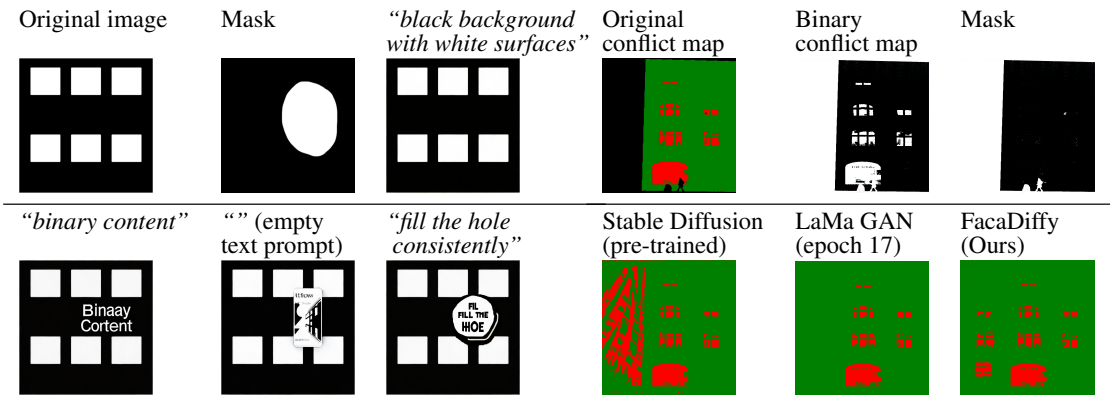

Figure 6. Examplary inpainting results demonstrating the strong dependence towards the text prompts.

Symmetry analysis. To analyze symmetry, we define a symmetry axis that intersects with the center of the image. This axis divides the image into two halves of equal size. After resampling, we compare the two halves by evaluating the Structural Similarity Index (SSIM). By iteratively rotating the axis and evaluating the similarity of both image halves, we approximate the best-fitting symmetry axis that intersects with the center of the image. We establish a criterion for the determination of the presence of symmetry, summarized in equation 1. We utilize the implementation of the SSIM available in the scikit-image Python library (van der Walt et al., 2014).

$$\text{Sym.} = \begin{cases} \text{True}, & \text{if } (I_{\text{max}} - \bar{I}) > \sigma_I \ \wedge \ I_{\text{max}} > t \\ \text{True}, & \text{if } (I_{\text{max}} - \bar{I}) > 3 \cdot \sigma_I \ \wedge \ I_{\text{max}} < t \\ \text{False}, & \text{otherwise} \end{cases} \tag{1}$$

With the maximal SSIM score I_{max}, the mean value of the SSIM scores $\bar{I}$, the standard deviation of the SSIM scores σ_I and the arbitrary threshold t.

Fragmentation analysis. To analyze the fragmentation of a conflict map, we first identify the contours with the corresponding OpenCV function (Bradski, 2000) that implements the contour detection algorithm of (Suzuki and Abe, 1985). In the subsequent step, we calculate the average and cumulative contour area and determine the fragmentation score according to equation 2. With the fragmentation score f, the mean contour area $\bar{A}$, the number of identified contours N, and the surface area of the n^{th} contour A_n.

Figure 7. Exemplary inpainting results on real conflict maps demonstrating the variability of the inpainting results when considering large masks. Before the inpainting process, the original conflict map was transformed into a black-and-white representation. The results were color-coded afterward.

$$f = \frac{\bar{A}}{(\sum_{n=0}^{N} A_n)} \tag{2}$$

Text prompt construction. We construct the text prompt from a set of pre-defined strings summarized in Equations 3, 4, and 5 according to the results of the previously discussed analyses. n_{black} and n_{white} represent the percentage of black and white pixels in the image. The fragmentation score f is assessed against a threshold $t_f = 0.98$ to discriminate the patch size. The symmetry of the image is evaluated according to the criterion established in Equation 1. We set the arbitrary threshold $t = 0.5$ and combine the strings 1, 2, and 3 to form the final text prompt.

4. Counterintuitive IoU Evaluation Resuls

When deploying the baseline methods, greater overlap due to larger continuous areas being inpainted into the images may positively impact the IoU, even though the underlying true semantic similarity would not be positively affected. Such counterintuitive results are exemplarily illustrated in Figure 9, where higher IoU measurements can be observed in examples where the true semantic similarity is obviously lower. The IoU measurements contradict the LPIPS and SSIM values which indicate the true, enhanced, semantic similarity that is obtained using FacaDiffy.

In contrast to evaluating the IoU instance-wise, we evaluate the entire conflict maps, which might also affect the score. Additionally, as (Zhang et al., 2018) mentions, no universal metric for quantifying image completion is available, which motivates us to consider multiple metrics for an unbiased comparison.

5. Additional Visualizations of Inpainting Results

Results on large masks. The inpainting examples on a real conflict map in Table 7 illustrate the performance of the different models considering larger masks.

Results on the CMP database. Figure 8 illustrates the inpainting results on a conflict map derived from the CMP database of annotated images.

References

Biljecki, F., Ledoux, H., Stoter, J., 2016. Generation of multi-LOD 3D city models in CityGML with the procedual modelling engine Random3Dcity. *ISPRS - Int. Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, IV-4/W1, 51–59.

Bradski, G., 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*. https://opencv.org/. Accessed: 2025-10-02.

Mariga, L., 2022. pyRANSAC-3D.

Patil, S., Cuenca, P., Kozin, V., 2022. Training stable diffusion with Dreambooth using diffusers. https://huggingface.co/blog/dreambooth. Accessed: 2024-01-11.

Radim, T., Radim, Š., 2013. Spatial pattern templates for recognition of objects with regular structure. J. Weickert, M. Hein, B. Schiele (eds), *Pattern Recognition*, 8142, Springer, Berlin, Heidelberg, Berlin, Heidelberg, 364–374.

Suzuki, S., Abe, K., 1985. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1), 32–46.

Tuttas, S., Stilla, U., 2013. Reconstruction of façades in point clouds from multi aspect oblique ALS. *ISPRS - Int. Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, II-3/W3, 91–96.

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., the scikit-image contributors, 2014. scikit-image: image processing in Python. *PeerJ*, 2, e453. https://doi.org/10.7717/peerj.453.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., Wang, O., 2018. The unreasonable effectiveness of deep features as a perceptual metric. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhou, Q.-Y., Park, J., Koltun, V., 2018. Open3D: A modern library for 3D data processing. *arXiv preprints*.

Figure 8. Inpainting results on an exemplary conflict map derived from the CMP database of annotated images. The inpainting was conducted in black and white, and the results were manually color-coded after the inapinting was completed.

String 1 =

$$\begin{cases} \textit{“an sks black background”} & \text{if } n_{\text{black}} > 50\% \\ \textit{“an sks white background”} & \text{if } n_{\text{white}} > 50\% \end{cases} \tag{3}$$

String 2 =

$$\begin{cases} \textit{“with large black patches”} & \text{if } f > t_f \ \wedge \ n_{\text{white}} > 50\% \\ \textit{“with large white patches”} & \text{if } f > t_f \ \wedge \ n_{\text{black}} > 50\% \\ \textit{“with small black patches”} & \text{if } f < t_f \ \wedge \ n_{\text{white}} > 50\% \\ \textit{“with small white patches”} & \text{if } f < t_f \ \wedge \ n_{\text{black}} > 50\% \end{cases} \tag{4}$$

String 3 =

$$\begin{cases} \textit{“that are consistent and symmetric with the rest of the image”} & \text{if symmetry} = \text{True} \\ \text{“ ”} & \text{if symmetry} = \text{False} \end{cases} \tag{5}$$

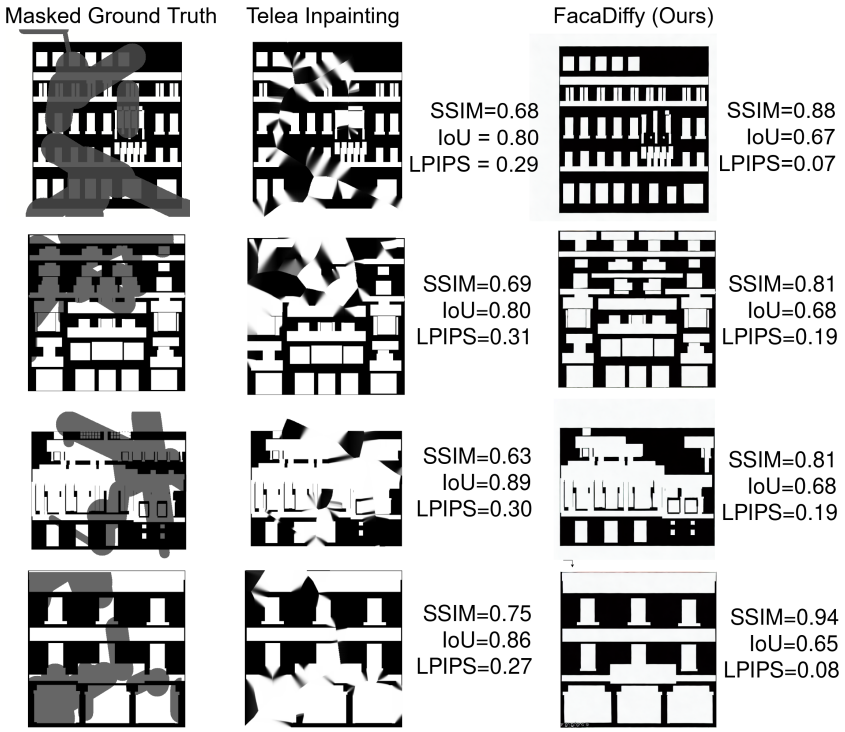

Figure 9. Exemplary inpainting results obtained with the traditional Telea inpainting method based on solving partial differential equations (PDEs) and with FacaDiffy. Semantic inconsistencies can be observed in the results obtained using the Telea Method. The IoU measurements, however, would, in contrast to the SSIM and LPIPS evaluation, suggest a larger similarity of these semantically inconsistent results to the ground truth.