Spot the Difference: Learned DSM Updating

BINGXIN KE[1], CORINNE STUCKER[1] & KONRAD SCHINDLER[1]

Abstract: In a rapidly changing world, keeping 3D city models up to date is a crucial task for many applications. Very high-resolution (VHR) optical satellite sensors, stereo matching algorithms, and learning-based refinement enable the reconstruction of high-quality city models from space, but so far did not account for the evolution of urban scenes. In order to rapidly update an existing city model with a limited set of newly collected satellite images, we introduce a DSM updating method based on neural implicit occupancy fields. We demonstrate that our method is able to effectively combine the old DSM and the new data. Consequently, changed areas can be updated to match new image observations, thus reducing the associated height errors (MAE) by ≈50% compared to the obsolete DSM; while the reconstruction quality in unchanged areas is ensured by the old DSM that is based on more imagery and thus a higher 3D point density.

1 Introduction

Reconstructing and maintaining an up-to-date city-scale scene is a fundamental and important task of photogrammetry and computer vision. The reconstructed high-resolution digital surface model (DSM) serves as a basis for various downstream applications, including topographic mapping, "digital twins", environmental simulations, planning, etc. Nowadays, very high-resolution satellite sensors enable the acquisition of optical remote sensing images with fine-grained details at almost any location on the Earth from different viewpoints in space, within a short time interval. The 3D reconstruction from remote sensing images is possible with the help of tailored stereo matching algorithms (ROTHERMEL et al., 2012; YOUSSEFI et al., 2020).

Due to limitations in terms of image resolution, geometric conditions, and radiometric consistency, the derived point clouds and DSMs tend to be noisy and sometimes incomplete. Recently, learning-based methods have been developed to refine the raw DSM from stereo reconstruction (BITTNER et al. 2018; STUCKER & SCHINDLER, 2022). We build on the *ImpliCity* method (STUCKER et al. 2022), which directly converts raw point clouds into a city model with smooth surfaces, fine-grained shape details, and crisp building edges.

Such existing methods aim to reconstruct DSMs without considering the acquisition time of the input images, and thus without regard for the evolution of the city over time. In actual fact cities do change, mainly due to construction activities. Still, most of the area in an urban scene remains static over a relatively long period, hence an old city model is expected to provide additional data redundancy in unchanged areas, especially in the practical scenario where one aims to keep the model up to date, and therefore update shortly after the change, when only few new satellite images have been collected.

[1] ETH Zurich, Photogrammetry and Remote Sensing, Stefano-Franscini-Platz 5, CH-8093 Zurich, E-Mail: [bingke, stuckerc, schindler]@ethz.ch

 DOI: 10.24407/KXP:1841069396

In this work, we propose a DSM updating method based on satellite images, using a neural implicit surface representation. With the help of a change detection module and properly pretrained encoders and decoder, the proposed method can leverage the existing DSM and newly collected data together to produce high-quality reconstructions of both changed and unchanged areas.

2 Methodology

2.1 Method Overview

Problem Formulation. The old DSM D_1 can be derived from a stack of satellite images $\boldsymbol{I}_1^{raw}$ that represent the state at time T_1, with conventional semi-global matching (ROTHERMEL et al., 2012) followed by surface reconstruction (STUCKER et al. 2022), as shown in the left half of Figure 1. Our goal is to update the old DSM state D_1 to a new state D_2, given only few (≥ 2) newly collected satellite images $\boldsymbol{I}_2^{raw}$ that represent the new state at time T_2. The updated DSM D_2 is expected to be up to date in the changed areas, but still have the same quality in unchanged areas.

DSM updating method. We decompose the problem into two sub-tasks: (1) determining changed areas; (2) fusing data and reconstructing the DSM by making use of all data in the unchanged areas, while ignoring the old DSM in changed areas. As shown in the right half of Figure 1, a new point cloud P_2 representing the new state is derived from the newly collected images $\boldsymbol{I}_2^{raw}$. Our method then takes as input the old DSM D_1, the point $\boldsymbol{P}_2$, and ortho-rectified images $I_1^{D_1}$ and $I_2^{D_1}$. The change detector g_ϕ takes one old state image $\mathrm{I}_1^{\mathrm{D}_1}$ and one new state image $\mathrm{I}_2^{\mathrm{D}_1}$ as input and outputs a binary change mask $\mathrm{M}_{\mathrm{change}}$ with pixel-wise change probabilities $\mathrm{p_i} \in \{0,1\}$, namely, $\mathrm{g}_\phi\left(\mathrm{I}_1^{\mathrm{D}_1}, \mathrm{I}_2^{\mathrm{D}_1}\right) = \mathrm{M}_{\mathrm{change}}$. Having this predicted change mask and other input data (old DSM D_1, new point cloud $\mathbf{P}_2$, and new state image pair $\boldsymbol{I}_2^{D_1}$), the DSM updater F_θ generates the new DSM D_2. Thus, our method can be formulated as:

$$F_\theta\left(\boldsymbol{P}_2, \boldsymbol{I}_2^{D_1}, D_1, g_\phi\left(I_1^{D_1}, I_2^{D_1}\right)\right) = D_2 \quad (1)$$

The change detector g_ϕ and the DSM updater F_θ are parameterized as deep neural networks. Note that due to the ortho-rectification of the imagery, D_1, M_{change}, $I_1^{D_1}$, and $I_2^{D_1}$ are inherently aligned in the same geographic coordinate system. I.e., the (x, y)-axes are the East and North directions in the local UTM zone, and the z-axis is the vertical.

2.2 Change Detector

Our change detector module is a Fully Convolutional Siamese architecture, following (DAUDT et al. 2018). Data is cropped into patches for training and inference. To relieve the bias caused by radiometric conditions, the patches are randomly rotated by $\alpha \in \{0°, 90°, 180°, 270°\}$ and randomly flipped along x or y axis during training. We further randomly swap the two input images to avoid asymmetries. Since it is in our context important to cover all changed content, even at the cost of including some unchanged parts, we trade precision for higher recall in post-processing: we perform erosion with radius $r = 10$ pixels (respectively, 2.5 m) to remove noise, followed by dilation with $r = 40$ pixels to cover slightly larger and more complete changed areas.

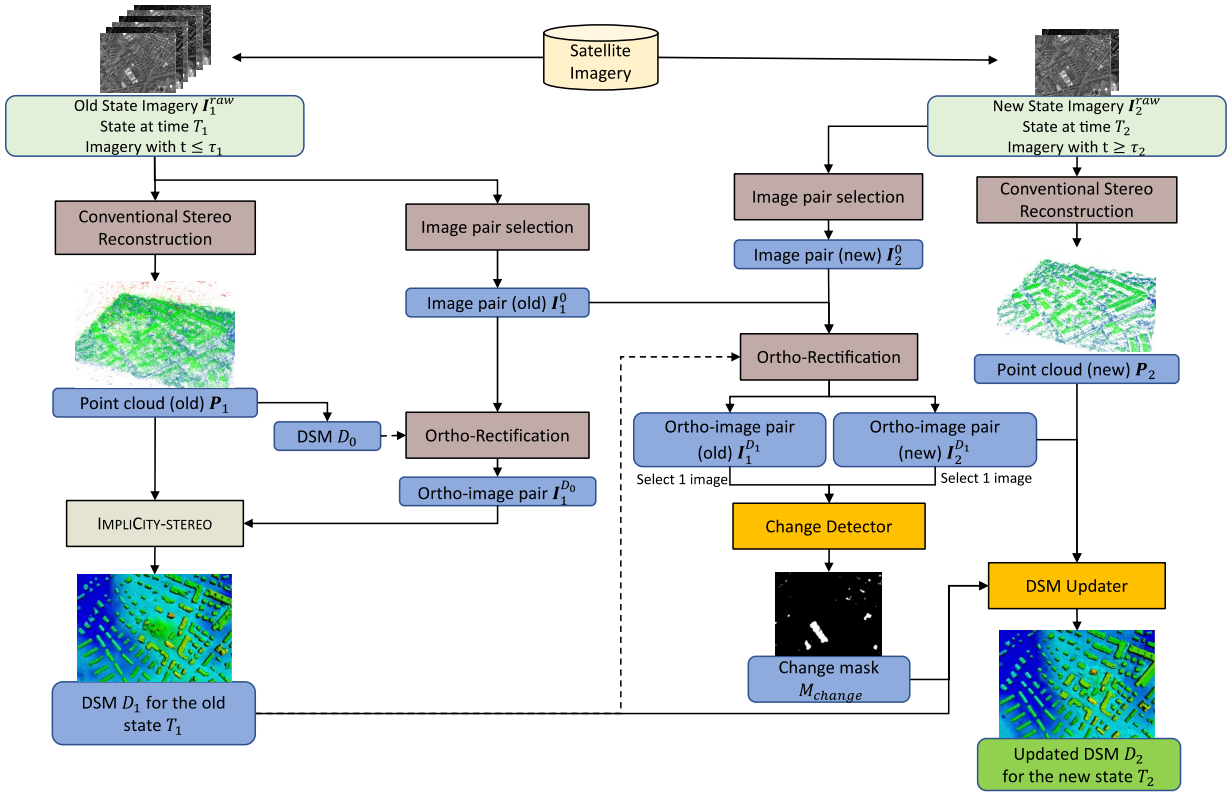

Fig. 1: Method overview

2.3 DSM Updater

The DSM updater module is the core of our method. It is adapted from the image-guided, coordinate-based neural representation of *ImpliCity*. The scene is represented as a convolutional neural occupancy field (Peng et al., 2020), i.e., a function f_θ that, for any given query point $x \in R^3$, returns the corresponding occupancy probability $\hat{o}$. In our case, $\hat{o} = 0$ for locations above the surface and $\hat{o} = 1$ for points underneath the surface.

Formulation. As shown in Figure 2, our network is an encoder-decoder design consisting of two raster encoders (DSM encoder f_{DSM} and image encoder f_{image}), one point cloud encoder f_{PC}, one feature fusion module f_{fusion}, and one decoder f_{decode}. The encoders (f_{DSM}, f_{image}, and f_{PC}) convert the input into feature embeddings. For a specific (x, y) location, the corresponding features and the change probability can be queried by bilinear interpolation from the embeddings. Then the occupancy value is predicted by the decoder. The prediction at a query point $\boldsymbol{x}$ can be written as:

$$f_\theta = f_{decode}\left(\boldsymbol{x}, f_{fusion}\left(f_{DSM}(D_1, \boldsymbol{x}), f_{PC}(\boldsymbol{P}_2, \boldsymbol{x}), M_{change}(\boldsymbol{x})\right), f_{image}\left(\boldsymbol{I}_2^{D_1}, \boldsymbol{x}\right)\right) \rightarrow \hat{o} \in [0,1] \qquad (2)$$

Network Components. The **point cloud encoder** follows the encoder architecture of *ConvONet* (Peng et al., 2020) and convert the input point cloud $\boldsymbol{P}_2$ into a high-dimensional feature embedding. The **image encoder** is the same as that in *ImpliCity*. It converts ortho-rectified images into an image feature embedding, to provide additional high-frequency details that would be missed in the sparser point cloud. The **DSM encoder** converts the old DSM into a feature embedding. It shares the same architecture as the image encoder, except for the first layer, which accepts single-

channel input. The **decoder** predicts the occupancy value from the sum of queried features (ψ_{fused} and ψ_{image}) and the coordinate of query point $\boldsymbol{x}$.

Feature Fusion. To smartly leverage the input data, we apply a feature-level fusion. Through ablation studies, we conclude that fusing the geometric features (i.e., DSM feature ψ_{DSM} and point cloud feature ψ_{PC}) yields the best result. Note that in changed areas (where $p = 1$), the DSM should be ignored, thus the fusion is disabled and ψ_{PC} replaces ψ_{fusion}.

Training. During training, query points $\{x_i \in R^3\}$ are randomly sampled within the volume of interest, with higher sampling density in the vicinity of building and terrain surfaces. The ground truth occupancy values are assigned according to the relative position of the sample point w.r.t. the ground truth DSM. The network is then trained by minimizing the cross-entropy loss $\mathcal{L}$ between predicted occupancies $\hat{o}$ and true occupancies o:

$$\mathcal{L}(\hat{\boldsymbol{o}}, \boldsymbol{o}) = \sum_i (o_i \cdot \log(\hat{o_i}) + (1 - o_i) \cdot \log(1 - \hat{o_i})) \tag{3}$$

Pretrained Weights. Using pretrained network modules yielded excellent reconstruction results in our ablation studies. Thus, in our final model, we use the pretrained weights to initialize the model and fine-tune them to the specific scene with a rather small learning rate.

DSM Generation. During inference, the updated DSM can be extracted from the neural occupancy field, as the iso-surface at $f_\theta = 0.5$. We do this in a coarse-to-fine search for efficiency. First, query points are coarsely sampled along the z-axis at every (x,y) raster cell of the output DSM. Then the z-range is recursively partitioned around the value $f_\theta = 0.5$ to find the precise surface height.

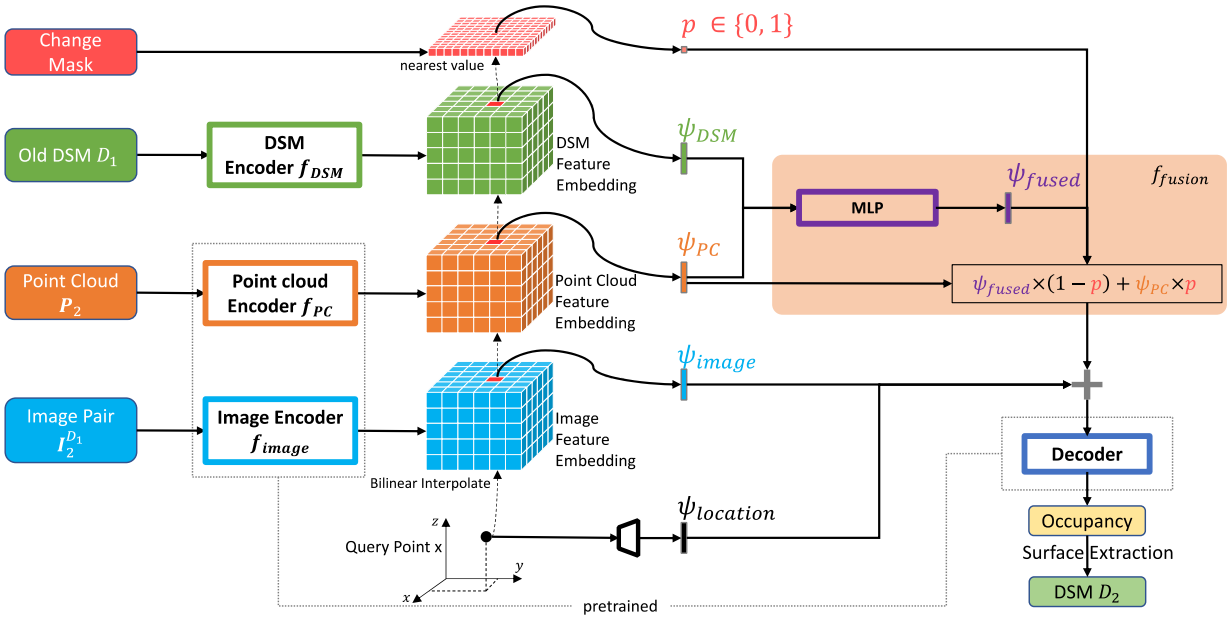

Fig. 2: Network architecture of the DSM updater

3 Experimental Setup

Dataset. We evaluate the proposed method on a satellite image dataset of WorldView-2 and WorldView-3 over Zurich, acquired between 2014 and 2018. The reference DSM is rasterized from the publicly available LoD2 city model of Zurich in 2015 and 2018. As shown in Figure 3, the study area consists of three rectangular sub-regions, covering various kinds of changed and unchanged buildings. The areas of these sub-regions are 1.89km^2 (ZUR_A), 1.84km^2 (ZUR_D), and 0.73km^2 (ZUR_C). By choosing 2017-09-01 and 2018-12-01 as time thresholds, as shown in Figure 4, we build up a typical map updating scenario, where we have several images from the past that represent the old state; and few images, collected recently within a short time interval, to capture the new state. The dataset is separated into three different geographical areas for training, validation, and testing.

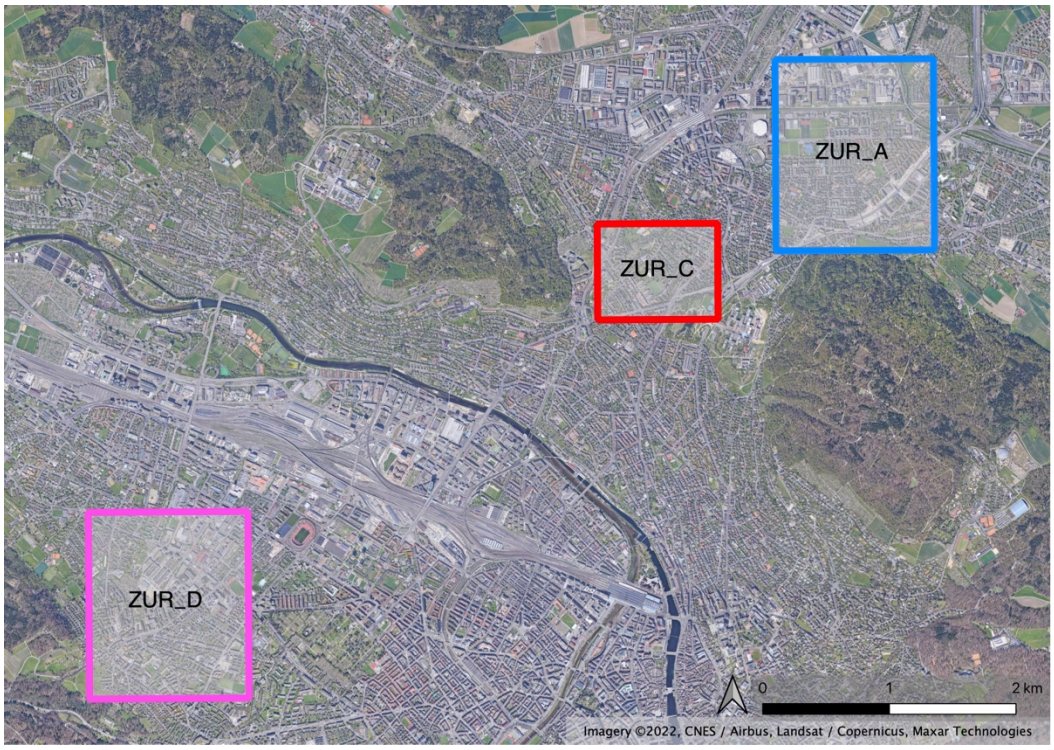

Fig. 3: Study area in Zurich. Three sub-regions with significant changes have been selected: ZUR_A is training area, ZUR_D is validation area, and ZUR_C is test area.

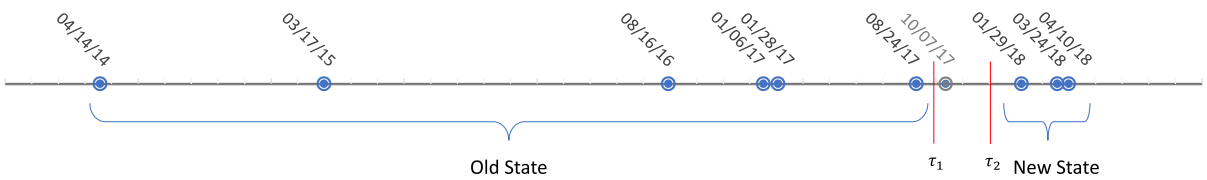

Fig. 4: Splitting of the image time series into data for the old and new states.

Baseline. We reconstruct the scene from scratch with only the new data by training the full *ImpliCity* network on the new data. This result represents the quality one can reach without taking into account any information about the old DSM state.
Evaluation Metrics. To quantify the quality of reconstructed and updated DSM, we calculate the mean absolute error (MAE) and the root mean square error (RMSE) over pixelwise deviations from the ground truth DSM.
Implementation Details. Our method is implemented in PyTorch and tested on a single NVIDIA GeForce RTX 2080Ti GPU. Training data is randomly sampled as patches with spatial dimension of 64m×64m and randomly augmented by rotations of $\alpha \in \{0°, 90°, 180°, 270°\}$ and flipping along the x- or y-axis. At inference time, we use a sliding window with 50% overlap to cover the whole validation or test area. Overlapping predictions are merged with linear blending.

4 Result and Discussion

Quantitative Results. Comparing the first two rows on Table 1 (i.e., old DSM vs. baseline), we see that both baseline reconstructions, using either only old data or only new data, have reasonably good quality. However, as expected the old data cannot correctly recover changed areas, whereas the new data alone suffers from low redundancy and yields worse overall performance. By comparing the 3rd and the 1st row, we see that leveraging the new data improves the MAE in changed area by $\approx 50\%$.
From the 3rd and the 2nd row we can see that the old DSM ensures the reconstruction quality in the unchanged area, and that the proposed selective DSM updating scheme outperforms the baselines in terms of overall statistics. The MAE drops by ≈ 0.2m and the RMSE drops by ≈ 0.3m.

Tab. 1: Quantitative comparison of reconstructed in the test area.

	Input state	Overall		Building		Terrain		Changed		Unchanged	
		MAE [m]	RMSE [m]	MAE [m]	RMSE [m]	MAE [m]	RMSE [m]	MAE [m]	RMSE [m]	MAE [m]	RMSE [m]
Input old DSM	Old	1.54	2.55	2.53	3.88	**1.23**	1.97	3.13	4.67	**1.50**	2.47
Baseline	New	1.71	2.69	2.45	3.87	1.48	2.22	**1.41**	**2.17**	1.71	2.71
Updated DSM	Old + New	**1.52**	**2.38**	**2.21**	**3.47**	1.31	**1.93**	1.59	2.63	1.52	**2.37**

Qualitative Results. Visual comparisons are shown in Figure 3. Firstly, we find that our method succeeds to update the content in changed areas from limited data, see white arrows in column (a) and (c). The demolished buildings are removed in the updated DSM, while the new buildings are reconstructed successfully.
Secondly, as pointed out by the violet arrows in column (a) and (b), our method correctly reproduces unchanged buildings that could not be reconstructed correctly only from the new data.
Moreover, as indicated by the red arrows in column (b) and (c), unchanged buildings that were missed in the old DSM can sometimes be recovered, too, when adding the new data. Moreover, the combined evidence tends to yield slightly sharper edges.

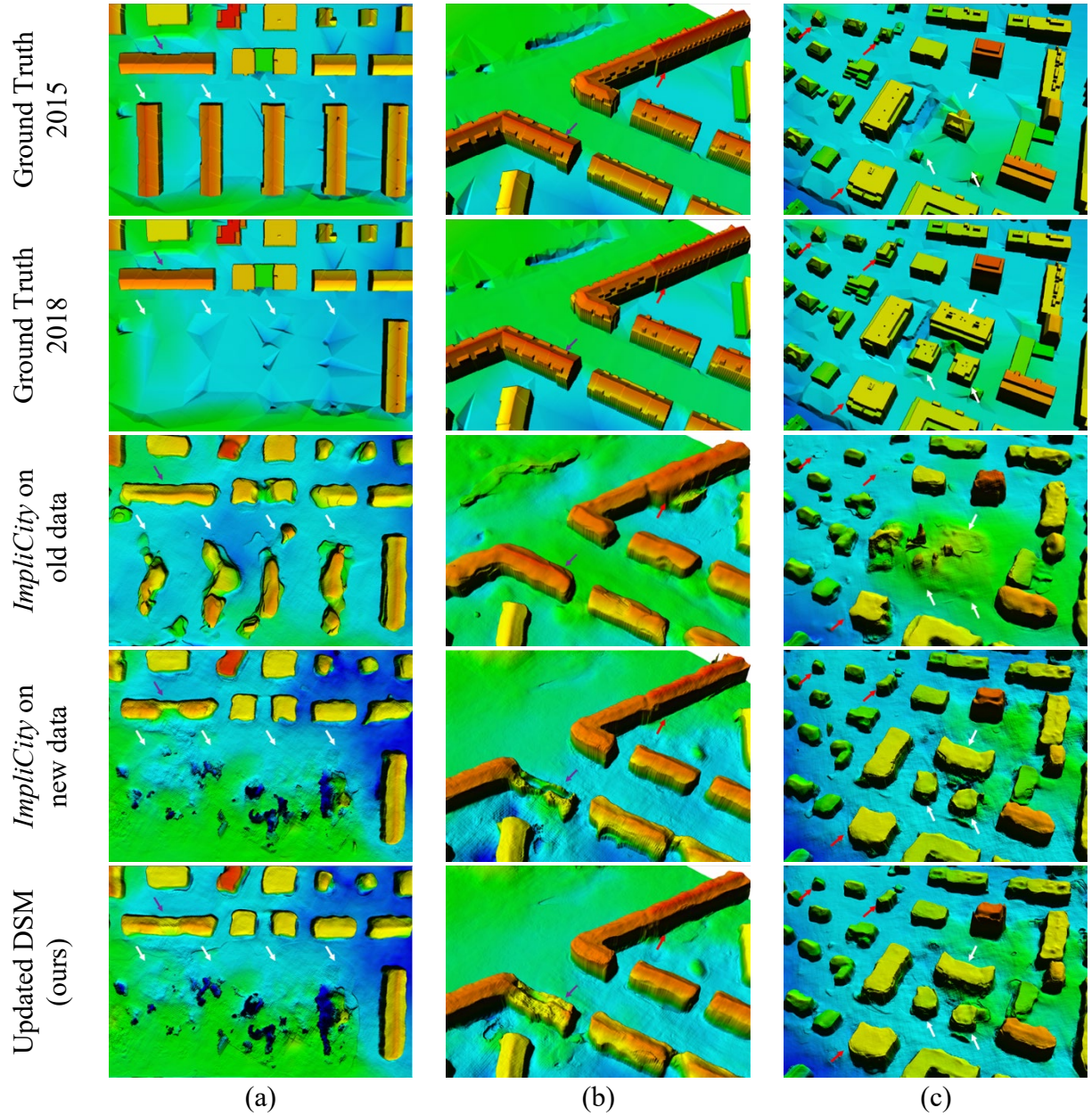

Fig. 3: Visual comparison of our result with the input DSM and the baseline in the test area. Heights are colored from blue to green to red. White arrows indicate the changed buildings; violet arrows indicate unchanged buildings that cannot be reconstructed as well from only new data; red arrows indicate unchanged buildings that are reconstructed better when using both old and new data.

5 Conclusion and Outlook

We have presented a DSM updating method for city scenes, using a DSM of an old, obsolete state together with satellite-based photogrammetric point clouds and ortho-photos generated with a small number of new images. The technical core of our method is a neural occupancy field. To the best of our knowledge, our work is the first learning-based approach for smart updating of city-scale DSMs based on satellite images. In summary, the conclusions of our study are:

1. *ImpliCity*, used from scratch with only little data that depict the new state, is able to reconstruct a DSM surprisingly well, still the quality is degraded in unchanged areas.
2. Leveraging the information captured in the old DSM together with the new data improves the reconstruction in the changed area without degrading unchanged areas. It is necessary to supply a reasonable change mask to guide the fusion, but that mask can be derived automatically from the available image data.
3. Suitably pretrained weights of the encoders and the decoder achieve comparable 3D quality in changed and unchanged areas, resulting in seamless fusion of old and new information.

In the wake of this exploratory study, potential improvements could be investigated in the future:

1. Reducing network complexity. The current method includes several fairly large modules. It would be desirable to reduce the parameter count while preserving performance.
2. Uncertainty guided data fusion. Introducing well-calibrated uncertainty estimation could provide additional information to better represent data dependencies.
3. Improving generalization. From an application point of view, it would be useful to ensure invariance w.r.t. varying imaging conditions, geographical areas, building styles, etc.

6 Acknowledgment.

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Contract #2021-21040700001. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

7 Bibliography

BITTNER, K., D'ANGELO, P., KÖRNER, M., AND REINARTZ, P., 2018: DSM-to-LoD2: Spaceborne Stereo Digital Surface Model Refinement. Remote Sensing, **10**(12), 1926, https://doi.org/10.3390/rs10121926.

DAUDT, R. C., LE SAUX, B. & BOULCH, A., 2018: Fully Convolutional Siamese Networks for Change Detection. Proc. ICIP, 4063-4067, https://doi.org/10.1109/ICIP.2018.8451652.

DE FRANCHIS, C., MEINHARDT-LLOPIS, E., MICHEL, J., MOREL, J.-M. & FACCIOLO, G., 2014: An Automatic and Modular Stereo Pipeline for Pushbroom Images. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., **II**-3, 49-56, https://doi.org/10.5194/isprsannals-II-3-49-2014.

PENG, S., NIEMEYER, M., MESCHEDER, L., POLLEFEYS, M. & GEIGER, A., 2020: Convolutional Occupancy Networks. Computer Vision – ECCV 2020, Vedaldi, A., Bischof, H., Brox, T. & Frahm, JM. (eds), Lecture Notes in Computer Science, **12348**, 523-540, Springer, Cham, https://doi.org/10.1007/978-3-030-58580-8_31.

ROTHERMEL, M., WENZEL, K., FRITSCH, D. & HAALA, N., 2012: SURE: Photogrammetric Surface Reconstruction from Imagery. Proc. LC3D Workshop.

STUCKER, C., KE, B., YUE, Y., HUANG, S., ARMENI, I. & SCHINDLER, K., 2022: ImpliCity: City Modeling from Satellite Images with Deep Implicit Occupancy Fields. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., **V**-2-2022, 193-201, https://doi.org/10.5194/isprs-annals-V-2-2022-193-2022.

STUCKER, C. & SCHINDLER, K., 2022: ResDepth: A Deep Residual Prior for 3D Reconstruction from High-resolution Satellite Images. ISPRS Journal of Photogrammetry and Remote Sensing, **183**, 560-580, https://doi.org/10.1016/j.isprsjprs.2021.11.009.

YOUSSEFI, D., MICHEL, J., SARRAZIN, E., BUFFE, F., COURNET, M., DELVIT, J.-M., L'HELGUEN, C., MELET, O., EMILIEN, A. & BOSMAN, J., 2020: CARS: A Photogrammetry Pipeline Using Dask Graphs to Construct a Global 3D Model. IGARSS 2020 - IEEE International Geoscience and Remote Sensing Symposium, 453-456, https://doi.org/10.1109/IGARSS39084.2020.9324020.