Revisiting the Necessity of Full Accuracy: Weakly Supervised Object-Level Offset Correction for Misaligned Building Labels

Supplementary Material

A. Multi-stage Alignment Framework Details

A.1. Self-supervised Alignment Algorithm

Our self-alignment algorithm is based on the observation of similarity between building edges and roof pixels. Given an initial building instance mask exhibiting positional misalignment, we utilize its shape characteristics and approximate spatial information to identify a local optimum that effectively aligns the mask with the target building via gradient descent.

A.1.1. Image Pyramid

To address the challenge posed by rich texture details in remote sensing images, we apply Gaussian blurring across multiple pyramid levels to suppress insignificant edges. At lower pyramid levels, larger Gaussian kernels are employed to preserve only prominent building edges, while assigning higher weights to edge consistency loss, thereby creating smoother loss landscapes. At higher pyramid levels, smaller kernels are used to retain more texture details, with reduced edge consistency weights, allowing variance consistency to guide the mask toward complete roof coverage. Fig. 1 illustrates the variations in mIoU between the offset estimations derived from different pyramid levels and the ground truth.

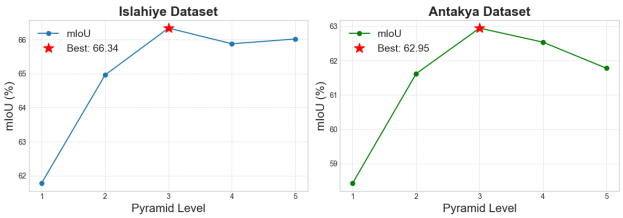


Figure 1. The mIoU of offset estimations across different pyramid levels.

Each pyramid level is optimized independently, except that higher levels are initialized using the results from lower levels. The number of pyramid layers was set to 3. Tab. 1 presents detailed parameter settings for both datasets.

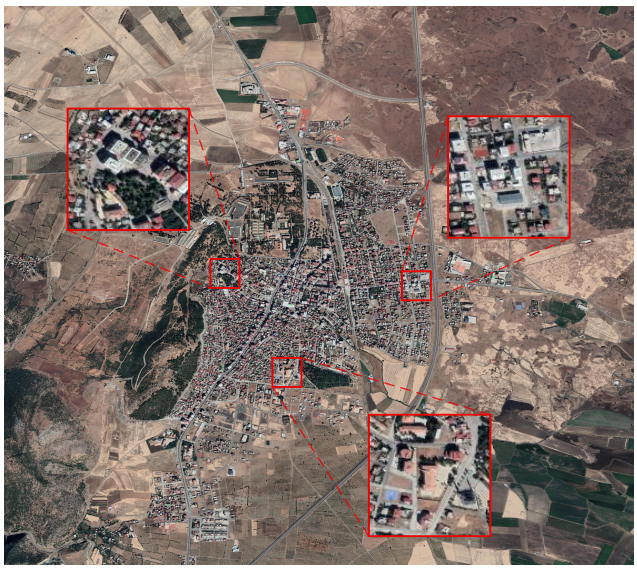


Figure 2. An example diagram illustrating the manual screening sampling process for the Islahiye dataset. The region enclosed by the red box represents the selected area requiring manual annotation.

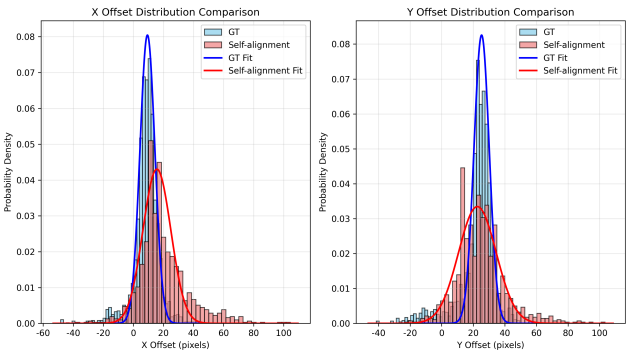
A.1.2. Differentiable Optimization

The key to making the optimization differentiable is the use of bilinear sampling to generate the translated mask $M_{\mathbf{v}}$. The value of the translated mask at a grid coordinate $\mathbf{p} = (x, y)$ is obtained by sampling from the raw mask M_{raw} at the continuously-valued coordinate $\mathbf{p}' = \mathbf{p} - \mathbf{v} = (x - dx, y - dy)$.

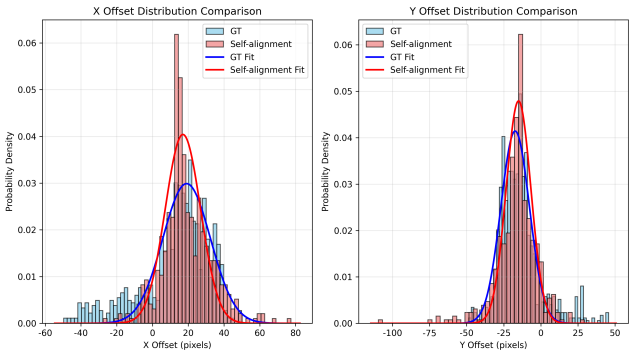
Let $\mathbf{p}' = (x', y')$. The value $M_{\mathbf{v}}(\mathbf{p})$ is interpolated from the four nearest integer grid cells in M_{raw}. Let $x_1 = \lfloor x' \rfloor$, $y_1 = \lfloor y' \rfloor$, and let $Q_{ij} = M_{raw}(x_1 + i, y_1 + j)$ for $i, j \in \{0, 1\}$. Let the interpolation weights for the horizontal and

Table 1. Hyperparameter settings for the 3-level optimization pyramid ($K = 3$). Each level k is configured with specific parameters for Gaussian blurring (σ, kernel size), optimization (iterations, learning rate η), and loss weighting (λ_{reg} for L_2 regularization and λ_{var} for variance consistency).

Level (k)	Name	Gaussian σ	Kernel Size	Iterations	Learning Rate (η)	λ_{reg}	λ_{var}
1	Coarse	20.0	41×41	50	2×10^{-3}	0.3	0.1
2	Medium	10.0	21×21	50	5×10^{-4}	0.7	0.3
3	Fine	4.0	9×9	50	1×10^{-4}	0.7	1.0



(a) Offset distribution in the Islahiye dataset



(b) Offset distribution in the Antakya dataset

Figure 3. Comparison of offset distributions between manually annotated labels (GT) and our self-alignment algorithm. The X and Y axes represent horizontal and vertical offsets respectively. (a) Islahiye dataset; (b) Antakya dataset.

vertical axes be defined as:

$$w_x(i) = 1 - |x' - (x_1 + i)| \tag{1}$$

$$w_y(j) = 1 - |y' - (y_1 + j)| \tag{2}$$

The interpolated value $M_{\mathbf{v}}(\mathbf{p})$ is then given by:

$$M_{\mathbf{v}}(\mathbf{p}) = \sum_{i,j \in \{0,1\}} Q_{ij}\, w_x(i)\, w_y(j) \tag{3}$$

The loss functions $\mathcal{L}_{edge}$ and $\mathcal{L}_{var}$ are functions of this generated mask $M_{\mathbf{v}}$. To find their gradients with respect to $\mathbf{v}$, we use the chain rule. The crucial component is the partial derivative of $M_{\mathbf{v}}(\mathbf{p})$ with respect to the components of $\mathbf{v}$, dx and dy:

$$\frac{\partial M_{\mathbf{v}}(\mathbf{p})}{\partial dx} = \frac{\partial M_{\mathbf{v}}(\mathbf{p})}{\partial x'} \frac{\partial x'}{\partial dx} = \frac{\partial M_{\mathbf{v}}(\mathbf{p})}{\partial x'}(-1) \tag{4}$$

$$\frac{\partial M_{\mathbf{v}}(\mathbf{p})}{\partial dy} = \frac{\partial M_{\mathbf{v}}(\mathbf{p})}{\partial y'} \frac{\partial y'}{\partial dy} = \frac{\partial M_{\mathbf{v}}(\mathbf{p})}{\partial y'}(-1) \tag{5}$$

The derivatives $\frac{\partial M_{\mathbf{v}}(\mathbf{p})}{\partial x'}$ and $\frac{\partial M_{\mathbf{v}}(\mathbf{p})}{\partial y'}$ are the definitions of bilinear interpolation of the gradient of M_{raw} at location (x', y'). Thus, we can write:

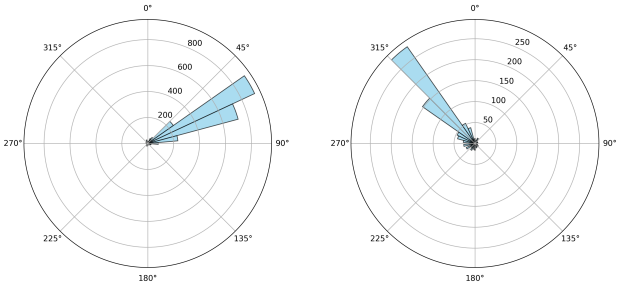
$$\begin{aligned} \nabla_{\mathbf{v}} M_{\mathbf{v}}(\mathbf{p}) &= \begin{pmatrix} \frac{\partial M_{\mathbf{v}}(\mathbf{p})}{\partial dx} \\ \frac{\partial M_{\mathbf{v}}(\mathbf{p})}{\partial dy} \end{pmatrix} \\ &= -\begin{pmatrix} \nabla_x M_{raw}(\mathbf{p} - \mathbf{v}) \\ \nabla_y M_{raw}(\mathbf{p} - \mathbf{v}) \end{pmatrix} \\ &= -\nabla M_{raw}(\mathbf{p} - \mathbf{v}) \end{aligned} \tag{6}$$

where $\nabla M_{raw}(\mathbf{p} - \mathbf{v})$ is the spatial gradient of the raw mask M_{raw} evaluated at the sampling coordinate $\mathbf{p} - \mathbf{v}$. This shows that the gradient with respect to the translation offset is the negative of the input mask's spatial gradient.

A.2. Prior-based Regularization

This component entails minimal human intervention, requiring only the selection of representative building regions that meet statistical criteria. Fig. 2 illustrates the sampling regions used in the Islahiye dataset. Offset statistics are computed via a sliding window approach (256×256 window, 128 stride) across 1024×1024 regions, effectively reducing errors arising from missing or inaccurate annotations.

Fig. 3 presents a comparison of offset distributions derived from manual annotations and those generated by our self-alignment algorithm for both datasets. The KL divergence between their respective 2D Gaussian fits is 0.56 for Islahiye and 0.26 for Antakya, indicating that our unsupervised alignment method produces corrections that are statistically comparable to manual annotations.



(a) Offset direction frequency in the Islahiye dataset

(b) Offset direction frequency in the Antakya dataset

Figure 4. Comparison of offset directions between the two datasets. (a) Islahiye dataset; (b) Antakya dataset.

B. Robustness Analysis of Self-Alignment Algorithm

Our unsupervised self-alignment algorithm's performance depends on dataset characteristics. To validate its robustness,

we carefully constructed the Islahiye and Antakya datasets with significant differences in offset patterns and label quality.

As shown in Fig. 4, the average offset directions in the two datasets are nearly orthogonal, demonstrating that our method exhibits no directional dependency. Fig. 5 provides a visual comparison of original label quality, revealing that the Antakya dataset contains more adherent labels with building instances often merged together, while the Islahiye dataset features more precise, well-separated annotations.

We found that the quality of the original offset labels has a significant impact on the accuracy of the offset estimation. This explains why our method achieves slightly lower performance on the Antakya dataset compared to Islahiye, as the adherent labels in Antakya present greater challenges for precise alignment. Despite these challenges, our method maintains robust performance across both datasets, handling varying offset patterns and label quality conditions effectively.

(a) Original Images (b) Ground Truth (c) Offset Labels

Figure 5. A comparison of the morphological forms of the original offset labels. The first row shows the Islahiye dataset, and the second row shows the Antakya dataset.

C. Comparison with Alternative Label Correction Methods

Many existing label correction methods [1, 4, 5, 11] rely on either artificially generated misalignments for training or extensive manual annotation, both of which conflict with our objective of rapidly augmenting data to improve model generalization. Other approaches either assume that the initial labels are largely correct [3] or suffer from poor reproducibility due to unavailable pre-trained model weights and source code [2, 7, 8]. Consequently, to ensure a fair and transparent evaluation, we refrain from comparing our method with these approaches.

Instead, we establish a robust baseline by adapting the method [10], which estimates the spatial offset field by maximizing the cross-correlation between image gradients and building footprints. In addition, we evaluated two representative works from the natural scene image domain [6] and the medical image domain [9], respectively. Since these methods are not designed to address this type of 2D translational noise, they yield unsatisfactory performance on our task. A quantitative comparison between our proposed approach and this baseline is presented in Tab. 2. Our unsupervised self-alignment algorithm achieves significant improvements, outperforming the cross-correlation baseline by 5.05% and 7.85% in mIoU on the Islahiye and Antakya datasets, respectively. Furthermore, our OMAF, which incorporates a small amount of supervisory signal, further enhances performance, achieving gains of 12.03% and 10.27% over the baseline on the respective datasets.

Table 2. Performance comparison against other methods. Our methods show significant improvements in mIoU (%) on both datasets.

Method	**Islahiye (%)**	**Antakya (%)**
SC [9]	55.53	50.67
ADELE [6]	56.98	51.76
Cross-correlation [10]	61.29	54.55
Self-alignment	66.34	62.40
OMAF	**73.32**	**64.82**

References

[1] Nahian Ahmed, Rashedur M. Rahman, Mohammed Sarfaraz Gani Adnan, and Bayes Ahmed. Dense prediction of label noise for learning building extraction from aerial drone imagery. *International Journal of Remote Sensing*, 42(23): 8906–8929, 2021.

[2] Dimitri Bulatov. Alignment of building footprints using quasi-nadir aerial photography. In *Scandinavian Conference on Image Analysis*, pages 361–373. Springer, 2019. 3

[3] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Autocorrect: Deep inductive alignment of noisy geometric annotations. *arXiv preprint arXiv:1908.05263*, 2019. 3

[4] Nicolas Girard, Guillaume Charpiat, and Yuliya Tarabalka. Aligning and updating cadaster maps with aerial images by multi-task, multi-resolution deep learning. In *Asian Conference on Computer Vision*, pages 675–690. Springer, 2018. 3

[5] Nicolas Girard, Guillaume Charpiat, and Yuliya Tarabalka. Noisy supervision for correcting misaligned cadaster maps without perfect ground truth data. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 10103–10106. IEEE, 2019. 3

[6] Sheng Liu, Kangning Liu, Weicheng Zhu, Yiqiu Shen, and Carlos Fernandez-Granda. Adaptive early-learning correction for segmentation from noisy annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2606–2616, 2022.

[7] Volodymyr Mnih and Geoffrey E Hinton. Learning to label aerial images from noisy data. In *Proceedings of the 29th International conference on machine learning (ICML-12)*, pages 567–574, 2012. 3

[8] John E Vargas-Muñoz, Sylvain Lobry, Alexandre X Falcão, and Devis Tuia. Correcting rural building annotations in openstreetmap using convolutional neural networks. *ISPRS journal of photogrammetry and remote sensing*, 147:283–293, 2019. 3

[9] Jiachen Yao, Yikai Zhang, Songzhu Zheng, Mayank Goswami, Prateek Prasanna, and Chao Chen. Learning to segment from noisy annotations: A spatial correction approach, 2023.

[10] Jiangye Yuan and Anil M Cheriyadat. Learning to count buildings in diverse aerial scenes. In *Proceedings of the 22nd ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 271–280, 2014. 3

[11] Armand Zampieri, Guillaume Charpiat, Nicolas Girard, and Yuliya Tarabalka. Multimodal image alignment through a multiscale chain of neural networks with application to remote sensing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 657–673, 2018. 3