Hilbert Curve-Based Attention Enabling Topology-Preserving Image Tensor Representation for Semantic Segmentation Network

Supplementary Material

7. BD3-Seg: Construction of the Building Defect Segmentation Dataset

The defect segmentation dataset presented in this paper is constructed based on the original BD3 classification dataset (Building Defect Detection Dataset, BD3) [14]. We reconstruct BD3 into a new dataset tailored for semantic segmentation of building defects. BD3 was introduced by Kottari and Arjunan in 2024 to provide a representative visual benchmark for building surface defect recognition. The dataset consists of real-world photographs captured directly from building surfaces and covers a wide range of typical construction materials (e.g., stone, lime mortar) as well as diverse environmental conditions, including outdoor natural illumination, shadow occlusion, and surface aging. Since the images were collected without structural constraints and preserve their natural scene conditions, they exhibit substantial variability in texture patterns, surface deformation, scale, and contamination. However, BD3 was originally designed for classification tasks and therefore lacks pixel-level annotations. To address this limitation, we re-annotate the key categories at the pixel level and construct a high-quality semantic segmentation dataset that serves as a new benchmark for building defect segmentation research.

Based on the overall distribution characteristics of BD3 and the requirements of our task, we reorganize the category system into two major groups: material categories and defect categories. The material group includes two major substrate materials—Lime and Stone—while the defect group contains three representative structural defects: Fissures, Peel-Off, and Mosses. On this basis, we carefully selected 700 high-quality, structurally complete, and representative images from the original BD3 dataset to construct the segmentation dataset used in this study.

We then employed professional annotators to perform pixel-wise semantic labeling following a strict annotation protocol, which includes:

- **Fine-grained boundary annotation**: Thin crack structures and irregular peel-off boundaries are refined through multi-point pixel-level adjustments to ensure continuity and completeness.
- **Material–defect disentanglement**: Material labels (lime, stone) are not mutually exclusive with defect labels, allowing the model to learn the semantic relationship between material texture and defect occurrence.
- **Accurate delineation of complex textures**: For stone textures that may be misidentified as cracks, annotators rely on color gradients, texture orientation, and local structural cues to improve labeling consistency.

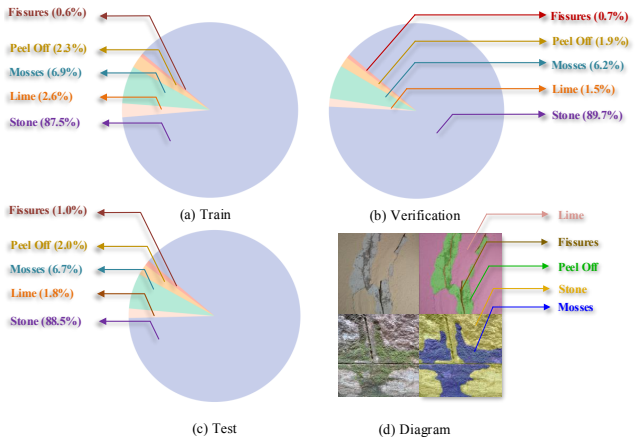

Figure 9. **Category Distribution of the BD3 Segmented Dataset.** The pie charts in (a)–(c) show the pixel-level category proportions for the training, validation, and test subsets, respectively, illustrating the highly imbalanced nature of the dataset where *Stone* dominates across all splits. Subclasses corresponding to building defects (*Fissures*, *Peel Off*, *Mosses*, and *Lime*) appear in much smaller proportions, reflecting the natural sparsity of real-world structural deterioration. Panel (d) presents representative annotated samples, demonstrating the visual appearance of each defect category and highlighting the complexity of fine-grained boundary structures on building surfaces.

After annotation, the 700 images were divided into training, validation, and test sets using a ratio of 5:1:1. The category distribution after splitting is shown in Fig. 9.

Across the three subsets, the category proportions shown in Fig. 9 remain highly consistent, indicating that the dataset was split in a random yet distribution-preserving manner. This consistency ensures reliable statistical behavior when training and evaluating models on different portions of the dataset.

The Stone class accounts for the majority of pixels (roughly 87%–90%), which reflects the sparse nature of defects on real building surfaces and results in a pronounced class imbalance. By contrast, classes such as Lime, Mosses, Peel-Off, and Fissures occupy much smaller regions. The particularly limited pixel counts for crack and peel-off regions underscore the difficulty of learning these fine-scale structures and highlight the need for methods capable of preserving detailed topology during segmentation.

The annotation examples in Fig. 9(d) show typical ap-

pearances of the defect categories: thin-line crack patterns, patch-like moss regions, localized surface loss in peel-off areas, and the textural variations among different materials. These characteristics illustrate the importance of accurate boundary localization and texture discrimination in building defect segmentation tasks.

8. Additional Discussion on Flattening Strategies

In this section, we analyze three spatial traversal strategies—Hilbert Curve, Z-Order Curve, and Row-First ordering—and provide the corresponding pseudocode used to compute their locality loss and runtime cost. We also include visualizations of these traversal patterns to highlight their differences in spatial smoothness and adjacency preservation, illustrating the structural advantages of Hilbert-based traversal.

Algorithm 1: Locality Loss Computation for Traversal Strategies

Input: Image resolution $H \times W$
Output: Locality losses $L_{\text{row}}, L_{\text{hilbert}}, L_{\text{z}}$
Function `RowFirst` (H, W): generate (x, y) in raster order;
Function `Hilbert` (H, W): generate (x, y) using Hilbert curve;
Function `Z-Order Curve` (H, W): generate (x, y) sorted by Morton index;

Function `LocalityLoss` $(coords)$:
  $loss \leftarrow 0$;
  for $i \leftarrow 0$ **to** $|coords| - 2$ **do**
    $(x_1, y_1) \leftarrow coords[i]$;
    $(x_2, y_2) \leftarrow coords[i+1]$;
    $loss \mathrel{+}= (x_1 - x_2)^2 + (y_1 - y_2)^2$;
  return $loss$;

Main Procedure:
  $C_{\text{row}} \leftarrow$ `RowFirst` (H, W) ;
  $C_{\text{hilbert}} \leftarrow$ `Hilbert` (H, W) ;
  $C_{\text{z}} \leftarrow$ `Z-Order Curve` (H, W) ;
  $L_{\text{row}} \leftarrow$ `LocalityLoss` (C_{row}) ;
  $L_{\text{hilbert}} \leftarrow$ `LocalityLoss` $(C_{hilbert})$;
  $L_{\text{z}} \leftarrow$ `LocalityLoss` (C_{z}) ;
  return $L_{\text{row}}, L_{\text{hilbert}}, L_{\text{z}}$;

8.1. Locality and Runtime Analysis

Row-First Traversal is the simplest and most commonly used flattening strategy. It scans an image line by line, ignoring spatial adjacency beyond horizontal continuity. As a result, Row-First generally exhibits the weakest topology preservation, since vertically adjacent pixels may be mapped far apart in the 1D sequence. This causes significant locality loss and negatively impacts the structural consistency of features during downstream processing.

Z-Order Traversal improves upon Row-First by recursively subdividing the image into quadrants and interleaving their coordinates. This hierarchical pattern strengthens locality preservation compared to Row-First, particularly for block-based structures. However, Z-Order still suffers from discontinuities across quadrant boundaries, and its overall continuity performance remains inferior to space-filling curves.

Hilbert Curve Traversal provides the strongest locality preservation among the three methods. As a continuous space-filling curve, the Hilbert Curve ensures that spatially adjacent pixels in 2D remain close in their 1D representations, minimizing locality disruptions. The curve effectively captures the topological structure of the entire image, which leads to lower locality loss and improved structural coherence in our method. This advantage is also reflected in our experimental results, where the Hilbert Curve achieves the lowest locality loss across all tested resolutions.

To complement the analysis in the main paper, this supplementary material presents the full pseudocode for computing the locality loss (Algorithm 1).

Algorithm 2: Computation of Runtime Cost for Three Traversal Strategies

Input: A set of image resolutions $\mathcal{S} = \{8, 16, 32, 64, 128\}$
Output: Runtime dictionary `results` storing execution time for each traversal
Initialize an empty dictionary `results`;
foreach $size \in \mathcal{S}$ **do**
  Initialize `results`$[size] \leftarrow \{\}$;
  `// Row-First traversal`
  Start timer;
  Call `RowFirstCoords`$(size, size)$;
  Record elapsed time in `results`$[size]$["Row-First"];
  `// Hilbert traversal`
  Start timer;
  Call `HilbertCoords`$(size, size)$;
  Record elapsed time in `results`$[size]$["Hilbert"];
  `// Z-order traversal`
  Start timer;
  Call `ZOrderCoords`$(size, size)$;
  Record elapsed time in `results`$[size]$["Z-Order"];
return `results`;

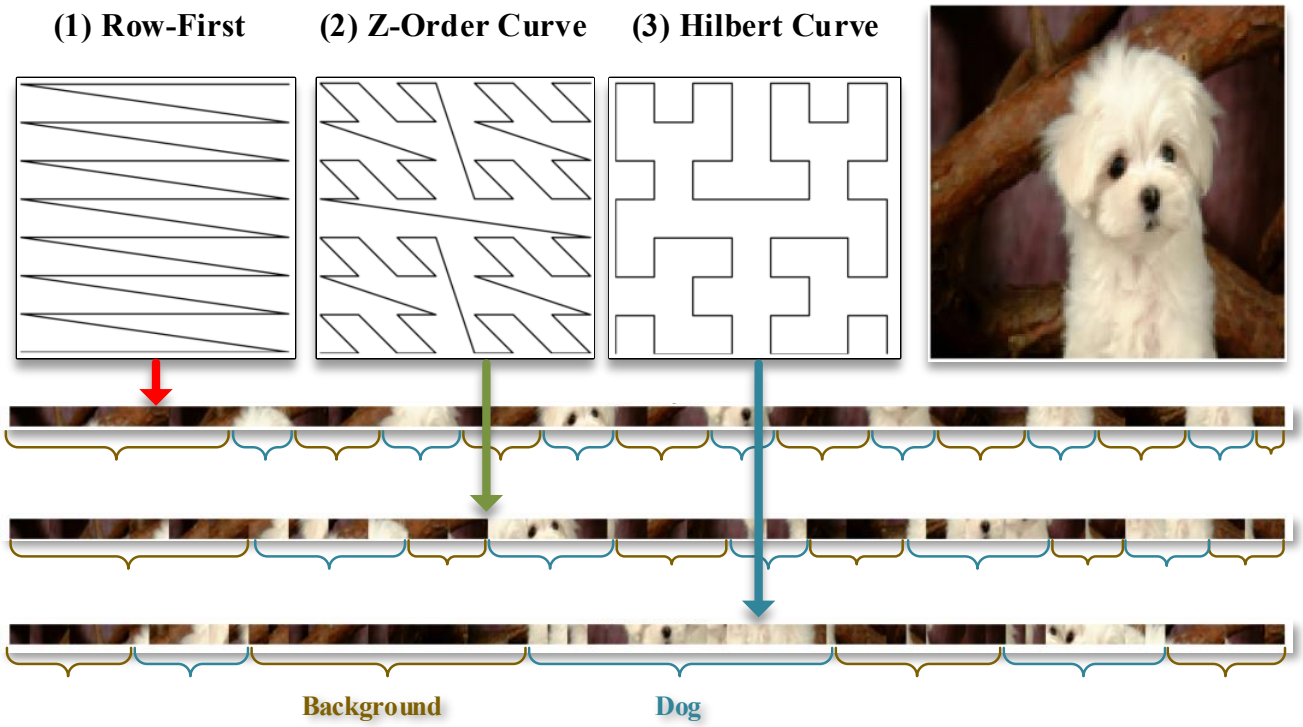

Figure 10. **Comparison of 2D-to-1D Flattening Patterns under Different Traversal Strategies.** The top row illustrates the traversal paths of Row-First, Z-Order, and Hilbert curves on a 2D grid, while the bottom sequences show how an example image (right) is reordered into 1D under each strategy. The rearranged pixel sequences reveal clear differences in locality preservation: (1) **Row-First** shuffles distant rows together, causing the *background* and *dog* regions to fragment into short, interleaved segments. (2) **Z-Order** better preserves small neighborhoods but still introduces abrupt cross-block jumps, resulting in medium-length segments with noticeable mixing at block boundaries. (3) **Hilbert Curve** maintains the strongest spatial coherence—pixels from the *dog* and *background* form long, continuous intervals with minimal interleaving. These visual patterns directly reflect each traversal's ability to preserve spatial adjacency, consistent with the locality-loss analysis presented in the main paper.

The locality loss serves as a quantitative metric to evaluate how well each traversal strategy preserves spatial adjacency after 2D–to–1D flattening, and it directly corresponds to the comparative results summarized in Table 1 of the main paper.

Algorithm 2 provides the full procedure used to measure the runtime cost of different traversal strategies. For each image resolution in the set $\mathcal{S} = 8, 16, 32, 64, 128$, the algorithm sequentially evaluates Row-First ordering, Hilbert curve traversal, and Z-Order traversal. The execution time of each method is recorded in a dictionary indexed by resolution and traversal type. These runtime measurements form the basis of the quantitative comparison reported in Table 2 of the main paper, where Hilbert traversal offers an effective balance between computational cost and structural continuity. Although Row-First remains the fastest due to its simple linear scan, it provides the weakest locality preservation. In contrast, Hilbert traversal maintains competitive runtime while achieving much stronger spatial adjacency preservation than both Row-First and Z-Order traversal.

8.2. Visual Analysis of Curve-Based Traversal Strategies

To further illustrate the structural differences among traversal strategies, Fig. 10 visualizes how Row-First, Z-Order, and Hilbert Curves transform a 2D image into a 1D sequence. The example image naturally separates into two semantic regions—background and dog —allowing a clear observation of how well each traversal maintains spatial coherence after flattening.

Row-First Traversal. The Row-First traversal produces long horizontal sweeps with abrupt resets at the end of each row. As shown in the visualization, these discontinuous jumps cause pixels from the background and dog regions—originally well separated in 2D—to become heavily interleaved in the 1D sequence. Instead of forming coherent contiguous blocks, the flattened representation alternates frequently between the two regions, especially at row boundaries where the traversal "cuts" across large se-

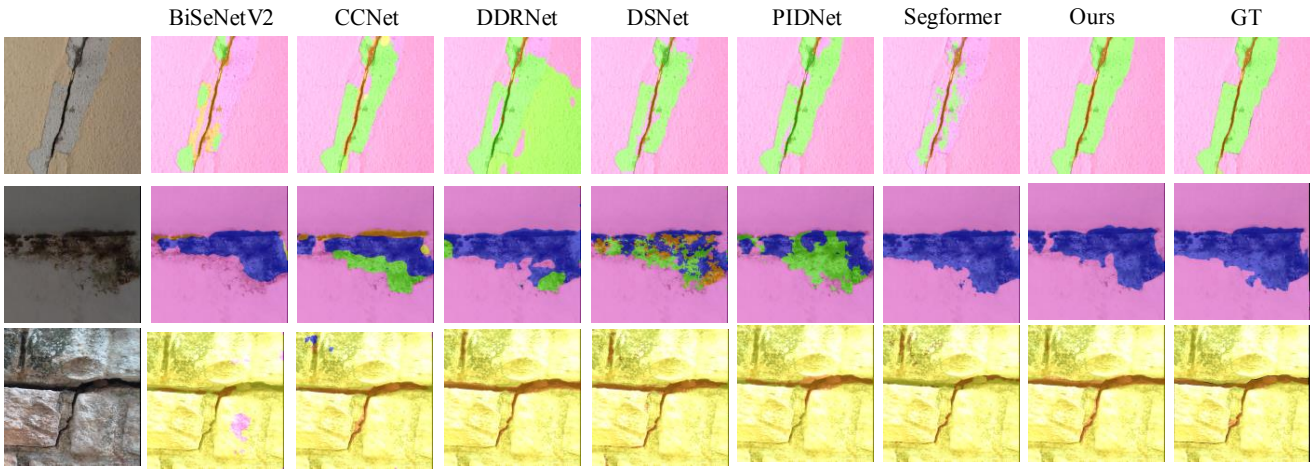

Figure 11. **Segmentation Results on the Validation Set.** Qualitative comparisons among different methods, including BiSeNetV2, CCNet, DDRNet, DSNet, PIDNet, SegFormer, and our approach. The samples cover multiple defect categories such as *Fissures*, *Peel-Off*, and *Mosses*, as well as complex background materials. Compared with existing methods, our model produces cleaner boundaries, more complete defect regions, and better topology preservation, closely matching the ground-truth annotations.

mantic gaps. This mixing demonstrates that Row-First disrupts spatial adjacency and offers weak locality preservation, making it unsuitable for tasks requiring consistent topological structure after flattening.

Z-Order Traversal. Z-Order traversal produces a distinctive block-based access pattern. Within each block, the curve preserves good local continuity, allowing neighboring pixels to remain relatively close after flattening. However, when the traversal moves from one block to another, it undergoes sharp directional flips, creating noticeable discontinuities in the resulting 1D sequence. As illustrated in the visualization, this leads to partially mixed distributions of background and dog regions: local patches remain coherent, but transitions between blocks introduce abrupt semantic switches that fragment the global structure. Consequently, Z-Order achieves moderate locality preservation—superior to Row-First due to its hierarchical subdivision, yet still prone to cross-block disruptions that weaken global spatial coherence.

Hilbert Curve Traversal. Hilbert traversal exhibits the most structurally coherent behavior among all strategies. The curve follows a smooth, continuous path that tightly preserves spatial adjacency, avoiding the abrupt directional jumps observed in Row-First and Z-Order. In the visualization, this continuity is reflected in the 1D sequence, where pixels belonging to the dog and background regions form long, uninterrupted segments rather than being repeatedly interleaved. Because the curve densely covers local neighborhoods before moving outward, even fine structures such as curved contours and soft texture transitions remain well preserved after flattening. This strong locality preservation enables Hilbert traversal to maintain the global shape integrity of semantic regions, making it the most topologically consistent method among the three.

9. Validation Set Visualization on BD3-Seg

To further illustrate the segmentation performance of the proposed method, this supplementary material provides qualitative visualizations on the validation subset. Fig. 11 presents representative samples containing different types of building defects, including Fissures, Peel-Off regions, and Mosses, as well as background material categories such as Lime and Stone. These examples highlight the challenges of fine-grained defect boundaries, complex surface textures, and the large intra-class variation present in the BD3 dataset.

As shown in Fig. 11, in the first row, DDRNet and PIDNet fail to accurately segment crack regions, while the other comparison methods can detect cracks but struggle to capture the surrounding peel-off areas. In contrast, TPSegformer produces results that are nearly identical to the ground truth. In the second row, CCNet, DSNet, and PIDNet exhibit significant segmentation errors on the moss category, with DDRNet also showing minor inaccuracies. In the last row, CCNet, DDRNet, DSNet, and SegFormer misclassify the joints between stone blocks as cracks, whereas TPSegformer also presents slight misclassification but to a much lesser extent. Overall, TPSegformer delivers segmentation results that are closest to the ground truth, with fewer mis-segmentations and more precise boundary delineation.

10. Experimental Results on the BD3-Seg Test Set

Table 8 provides the complete quantitative results on the BD3 test subset, including per-class IoU and overall metrics for all compared methods. These detailed measurements supplement the condensed results presented in the main paper and allow for a clearer comparison of the strengths and limitations of each segmentation network under the challenging conditions of the BD3 dataset.

Table 8. **Quantitative comparison on the BD3 test set.** All values are IoU (%). Best results are in bold.

Method	Fissures	Peel-Off	Mosses	Lime	Stone	mIoU	ACC
DSNet[9]	61.76	64.55	51.60	93.72	80.77	70.48	83.96
PIDNet[29]	60.29	81.77	66.31	95.01	79.92	76.66	**88.76**
CCNet[12]	59.65	82.11	64.02	94.90	78.90	75.92	85.14
SegFormer[28]	53.88	70.66	45.35	93.06	75.51	67.69	81.44
BiSeNetV2[31]	48.68	74.99	44.61	94.12	76.69	67.82	76.74
DDRNet[19]	50.68	53.38	48.94	89.23	79.11	64.27	85.88
Ours	**65.24**	**89.59**	**67.02**	**95.95**	**81.88**	**79.94**	88.33

Table 8 presents the quantitative comparison on the BD3 test set. Across all five categories—*Fissures*, *Peel-Off*, *Mosses*, *Lime*, and *Stone*—our method achieves the highest IoU, demonstrating consistently superior performance over all competing approaches. Specifically, the proposed model attains 65.24% on *Fissures*, 89.59% on *Peel-Off*, 67.02% on *Mosses*, 95.95% on *Lime*, and 81.88% on *Stone*, outperforming the second-best methods by clear margins in every category. These results highlight the effectiveness of our topology-preserving representation in handling both sparse, fine-grained defects and large-scale textured regions.

Furthermore, our approach achieves the best overall mIoU (79.94%) and ACC (88.33%), confirming strong generalization across the full spectrum of structural conditions. This comprehensive improvement aligns with the locality-preserving behavior analyzed in the main paper, where Hilbert-curve traversal delivers superior spatial continuity during feature mapping. Together, these findings demonstrate that the proposed method consistently delivers state-of-the-art performance across all semantic categories in the BD3 dataset.