WinSyn: A High Resolution Testbed for Synthetic Data

Tom Kelly[1] John Femiani[2] Peter Wonka[1]
[1]KAUST, KSA [2]Miami University, Ohio

Abstract

We present WinSyn, a unique dataset and testbed for creating high-quality synthetic data with procedural modeling techniques. The dataset contains high-resolution photographs of windows, selected from locations around the world, with 89,318 individual window crops showcasing diverse geometric and material characteristics. We evaluate a procedural model by training semantic segmentation networks on both synthetic and real images and then comparing their performances on a shared test set of real images. Specifically, we measure the difference in mean Intersection over Union (mIoU) and determine the effective number of real images to match synthetic data's training performance. We design a baseline procedural model as a benchmark and provide 21,290 synthetically generated images. By tuning the procedural model, key factors are identified which significantly influence the model's fidelity in replicating real-world scenarios. Importantly, we highlight the challenge of procedural modeling using current techniques, especially in their ability to replicate the spatial semantics of real-world scenarios. This insight is critical because of the potential of procedural models to bridge to hidden scene aspects such as depth, reflectivity, material properties, and lighting conditions.

1. Introduction

Larger and more sophisticated machine learning models demand an ever-increasing supply of data, particularly in tasks where manual annotation is challenging, such as depth estimation, reflectance estimation, or full 3D reconstruction. One solution is to use procedural models to generate synthetic training data. However, creating procedural models that accurately reflect the domain of real images (i.e. closing the domain gap) remains an open problem. Despite the visual realism of many synthetic scenes, their effectiveness in machine-learning applications often falls short. In this work we do not close the gap – but present an accessible pair of real and synthetic datasets, with annotations, in which this domain gap may be studied without massive resources.

While the final goal is to tackle complex problems, a straightforward proxy task is initially required to ensure that real-world imagery can be manually annotated and compared to synthetic imagery. We, therefore, propose segmentation as a proxy task. This proxy task helps pinpoint where a procedural model fails to capture the diversity and nuances of real-world scenes.

As our long-term goal is procedural urban modeling, we initially considered street-view images and datasets, such as CityScapes [12]. However, this has several drawbacks. Modeling arbitrary urban scenes realistically requires overwhelming complexity; requiring modeling at least cars, humans, buildings, skies, and vegetation, and each of these poses quite distinct and complex modeling challenges. For

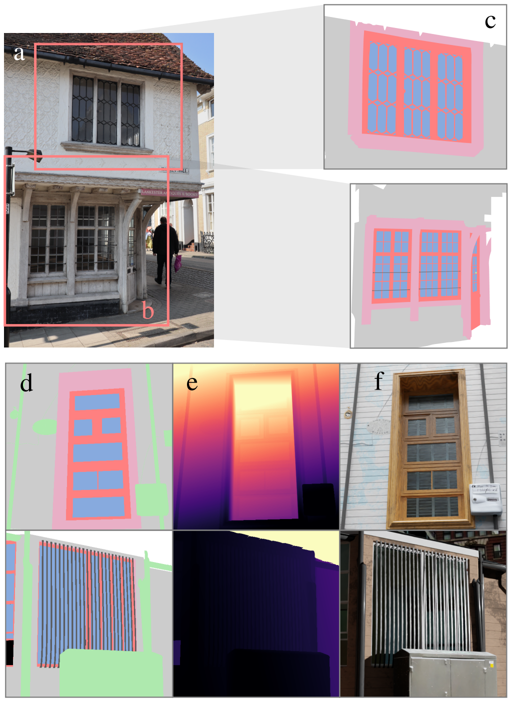

Figure 1. Photographers in 28 geographic regions captured real-world photos (a) of windows that are cropped (b) to single windows, which are then labeled (c). Synthetic windows are rendered giving color (f) and labels(d), while other passes such as depth (e) are also possible.

example, modeling urban environments for video games is a significant undertaking that often requires over 100 artists for open-world games. This type of effort is unrealistic for procedural modeling research. We propose that it is more promising to identify a simpler subset of images and develop a more constrained procedural model for this subset. This should enable faster iterations and broader participation in procedural modeling research, before scaling up to complete cities.

The decision to study the domain of window images was based on the following design principles for our dataset:

Diversity: Our goal is to capture a range of variations in both geometry and materials. We opted against humans due to the complexities in modeling realistic fine geometric details (like hair and beard) and materials (such as skin), and the limited topological variations. Although humans have been used for synthetic data studies as demonstrated in [39], exploration in this domain is complicated because the procedural model was not released and high-fidelity models such as Unreal Metahumans by Epic Games [15] tend to prioritize realism over variability. Plants, another option, present challenges in material diversity and are difficult for human annotators to segment. Windows, in contrast, offers a balanced combination of geometric complexity and material variation suitable for our research goals.

High-resolution: The images in the dataset should be high resolution to make the dataset useful for the near and medium-term future. This is in contrast to existing architectural datasets, with resolutions typically in the 512-1025 pixel range [5, 6, 25, 33]. While these are not low resolution in typical computer vision terms, architectural features such as windows are often too small to resolve in the images. Although the CityScapes dataset is larger than most, with a resolution of 2048×1024 pixels, it often captures architecture from oblique viewpoints limiting the effective resolution.

Image Rights: We would like to have all rights to the images to avoid future copyright problems.

To explore this problem, we introduce a specialized dataset that bridges real-world architectural imagery and procedurally-generated images, with a particular focus on window designs. Our dataset, comprising 89,318 windows in 75,739 high-resolution (4K to 6K) and RAW images, not only matches the scale of established datasets like CelebA-HQ and FFHQ but also offers a unique niche with an emphasis on architectural details and high resolutions.

Our dataset is designed not just to advance synthetic data generation, but also to facilitate diverse research avenues. Our primary contributions[1] are:

1. **Real-World Imagery:** A 4K resolution dataset with 75,739 photos of windows from global locations, offering unprecedented detail for architectural research.
2. **Hand-Annotated Labels:** Segmentation labels for 9,002 images.
3. **A Procedural Model:** A novel procedural model for windows, underscoring key design choices for synthetic data generation.
4. **Synthetic Data and Labels:** A diverse set of 21,290 synthetic window images, mirroring features observed in real imagery.

Additionally, we conduct extensive experiments and ablations to understand the impact of various features in our synthetic dataset on segmentation performance.

2. Related Work

Several datasets of architectural imagery have been created, enabling applications such as architectural style classification [2, 9, 40], building functional use classification [21, 41], architectural heritage classification [27], landmark identification [31], or urban scene matching [18]. Notably, datasets that support image synthesis or image segmentation have been instrumental in advancing these fields. For instance, the FaSyn13 dataset [13] consists of 200 façade images for texture synthesis, but its size is limited for modern generative models. Similarly, the LSAA dataset [42] includes 199,723 façade images and 516,000 cropped window images. However, the resolution of these cropped windows varies, with the majority being less than 100 pixels in the longest dimension.

In comparison, extensive collections like MS COCO [26] and LAION [35] offer a broader range but with challenges in resolution and scene variety. For instance, the sheer diversity in LAION's 9.8 million 1K-resolution images presents significant challenges for a single procedural model, highlighting the need for focused datasets. Our dataset addresses this by providing high-resolution window images, offering a specialized resource for detailed architectural analysis and procedural modeling.

Our windows-focused dataset, while containing fewer windows than LSAA, stands out as the highest-resolution dataset of window images known to us, with an average resolution of 4,000 pixels per side for cropped windows. This high-resolution focus, particularly in the 4K to 6K range, fills a unique niche in architectural and synthetic-to-real domain transfer research. Our dataset's specialization in high-resolution architectural elements, especially windows, offers a valuable resource for advancing procedural modeling, emphasizing the importance of detail and precision.

Several architectural image datasets supporting semantic segmentation or façade parsing exist, though not specifically focused on windows. The Graz dataset [33] contains 50 rectified images, and the eTRIMS datset [25] contains 60

[1] Available online https://twak.github.io/winsyn

non-rectified images. The CMP-Facade dataset [36] contains 606 images, with about half of them fairly high resolution (1,024 pixels on the long edge) but a limited diversity of image locations. The LabelMe-Facade dataset [5] has the largest number of images at 945, with each image varying in size between 512 and 768 pixels on a side. However, these datasets do not have the number of images nor the resolution that are desirable for training the latest computer vision methods, which increasingly require more detailed and high-resolution data. With 9,002 labeled images at four times the resolution of these datasets, our proposed dataset of real-world images is an order of magnitude larger. By concentrating exclusively on windows, our dataset offers unprecedented detail and specificity, enabling more precise and effective models for architectural element analysis.

Various authors have attempted to use synthetically generated data to bootstrap performance on real images. This approach seems to work best in domains where the human annotation is not directly feasible, such as reinforcement learning, especially for driving applications [14], depth or optical flow [8, 16], or 6DoF pose estimation for robotic grasping or manipulation [19, 24, 37].

Infinigen [32] is a good example of a procedural model, however, there is no validation of the effectiveness of the model for machine learning tasks. Of particular note is the SynthIA dataset [34], a driving dataset built on video game technology specifically designed to support semantic segmentation in urban environments. A very large engineering effort went into this, as well as CARLA [14], and we believe that reproducing such high-quality synthetic data is out of reach for most academic teams. Similar to our dataset, SynthIA aims at pushing the envelope to use synthetic data to improve computer vision even for problems where large human-annotated datasets (KITTI [28], LabelME-Facade [6], Camvid [4]) already exist. Similar to our findings, they can get some results from purely synthetic data, but they cannot out-compete even relatively small real-world labeled images, but by combining synthetic and real at a 4:6 ratio they obtain their best results. However, unlike SynthIA, our segmentation challenge is more constrained (only windows) and we think would require fewer resources for academic researchers to develop competing procedural models for synthetic training. Although buildings are visible in these datasets, they are not focused on architecture. Domain transfer for architecture is challenging due to the amount of variety and complex dependencies between architectural elements, whereas the categories of objects relevant in driving scenes are much more clearly defined. In addition, the high resolution of the images we use makes the synthesis of realistic textures and precise object boundaries critical, and the higher-capacity segmentation models of today vs. 2016 (when SynthIA was published) are more precise but may also be more likely to overfit synthetic data.

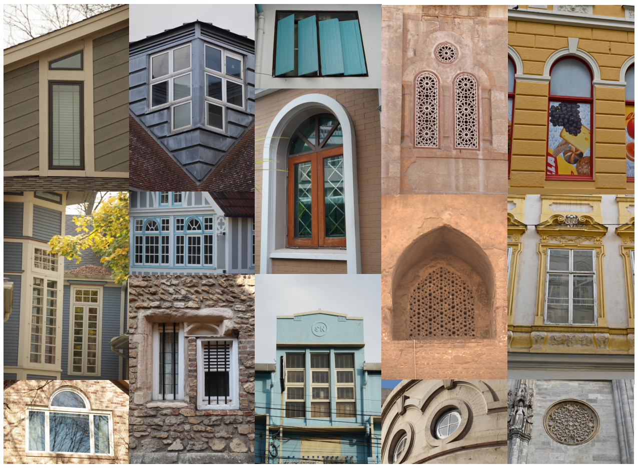

Figure 2. Samples from the 75,739 photographs in the dataset. Each column shows a variety of examples of windows from different geographic locations. From left to right: Chicago (USA), Cambridge (UK), Bangkok (Thailand), Cairo (Egypt), and Vienna (Austria). The dataset has a variety of window shapes and architectural styles.

Our dataset of real and synthetic imagery is unique as a high-resolution, voluminous dataset and serves as a proving ground for synthetic to real training, image generation, and semantic segmentation tasks.

To our knowledge, the most successful work on using synthetic data for segmentation is the ‘Fake It Till You Make It’ paper by Wood et al. [39], which reports improvements in segmentation when a U-Net is trained using their synthetic data vs. real image. However, to get these improved results, they used label adaptation, a technique that requires real labeled data. This is counter to our goals for using synthetic data for domains where labels are scarce. They do not report segmentation results without label adaptation, but they do show in their ablation study on landmark localization that label adaptation is critical for benefiting from synthetic data.

3. Real-Windows Dataset

In line with our goal of advancing procedural modeling, our dataset is curated to offer high-resolution, diverse imagery essential for developing and evaluating procedural models. Our dataset, comprising over 75,739 high-resolution images (up to $4K \times 6K$), is one of the largest and most detailed collections of window imagery available. Unlike other datasets sourced from the web or Flickr [22, 23], we hold complete copyright ownership of every image, ensuring legal clarity and flexibility for research use[2].

The dataset’s diversity in location, viewpoint, and archi-

[2] While the datasets we referenced use images under permissive licenses, owning our images outright simplifies usage rights.

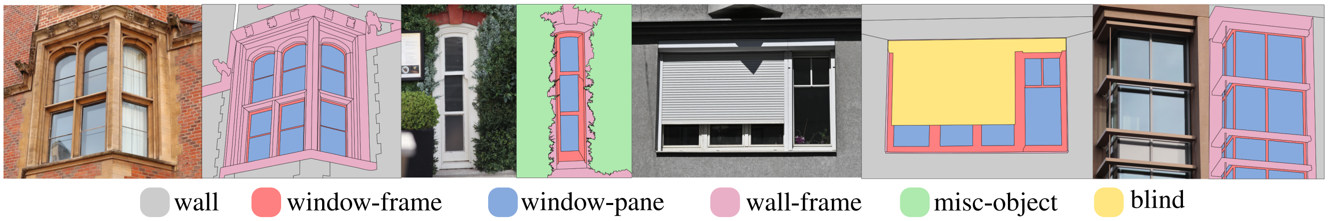

Figure 3. Examples of the labels used to annotate our data. Each instance receives its own polygon. The reader may wish to zoom into the figure for details.

tectural style, was achieved through a global effort. We engaged photographers from various countries, primarily hired via Upwork, to capture a wide array of window designs. This global collection effort ensures our dataset represents a broad spectrum of cultural and architectural diversity. This diversity is crucial for procedural models to learn and adapt to a wide range of architectural styles, directly supporting our goal of creating versatile and realistic models. Each image was chosen to highlight the window's design and architectural context, ensuring clear, distraction-free, and well-framed shots. The emphasis was on professional-grade photography with a focus on 4K resolution, achieving at least 2K pixels across each window, and taken during the day to ensure balanced exposure and minimal noise.

In curating this collection, we also prioritized ethical considerations, instructing photographers to avoid capturing private situations or sensitive locations. Over 12 months, this project involved hiring 30 photographers and incurring costs of US $0.20-$0.50 per image, in addition to our quality control and subcontract management expenses.

The diversity of image locations is indicated in Table 1 of the Appendix, along with the number of images that include semantic segmentation labels and RAW camera data. The RAW camera data's higher bit-depth could be particularly valuable for future procedural modeling research, offering richer information for model training and evaluation.

Many photos included multiple windows. We manually annotated crops in each image as a region that includes a single window in the center, along with any portion of the wall that may have been adapted to the window (such as brickwork or molding) and a portion of the wall on all four sides of the window. Due to this cropping, window images are in a variety of sizes, as shown in Appendix Fig. 2. We store the original images and the cropping information separately and generate a cropped version of the dataset on demand.

Each image has been carefully cropped and then annotated to focus on a single window and its immediate architectural context, as illustrated in Fig. 3. The annotations are designed to test if procedural models accurately replicate real-world architectural elements. The cropped window images also underwent a manual annotation process for panoptic segmentation, and subsequent review for quality assurance, leading to the acquisition of 9,002 annotations at an average cost of US $3.90 per image. While manageable, this cost notably exceeds image acquisition expenses, underscoring the value of methods that reduce reliance on labor-intensive labeled data for segmentation tasks. A detailed account of the annotation process is presented in Section 5.

Label	Images Using	Area %
wall	8907	43.02%
window pane	8362	22.58%
wall frame	8697	14.91%
window frame	8681	9.71%
unlabeled	2994	3.09%
shutter	931	2.56%
balcony	973	1.08%
misc object	2357	1.07%
blind	375	0.75%
bars	679	0.68%
open-window	977	0.55%

Table 1. The labels used, their frequency of use, and percentage by area for the square-cropped dataset used for our experiments. We note that the dataset is a mix of well-used labels such as wall and less-used ones, such as blind or bar. The 'unlabeled' category contained areas beyond the building (e.g., sky, streets) and a much smaller number of ambiguous areas where we could not reach a decision on how to label a feature.

The WinSyn dataset serves as a foundational step towards our larger goal of procedural urban modeling, offering a comprehensive and detailed resource for both current and future research in this domain.

4. Procedural Model Development

In developing our procedural model, we aim to create a realistic synthetic dataset with the ability to produce variations (ablations and experiments) for evaluation against real-world data. The compact domain (windows are relatively simple to model) allows well-developed approaches to be used to create the scene geometry including Split

Shape Grammars [38] and the CGA language [29], combined with Bézier splines for curved window shapes. This approach ensures that our model can generate a wide range of architectural styles and window designs closely resembling real-world distributions. The canonical orientation offered by windows as a domain allows a layered approach to geometry generation, as Appendix Fig.11: from the camera, we generate street clutter (e.g., cars, bollards, shrubs), wall decorations (shutters, balconies, pipes) exterior walls, window geometry, window dressing (blinds, curtains), and interior geometry.

The procedural modeling pipeline, developed in Python, Blender [11], and rendered with the physically based Cycles [3] renderer is used to generate a diverse dataset of 21,290 synthetic window images with corresponding labels (as Appendix Fig. 4). Normal, depth (Fig. 1), and edge maps are optionally generated. The model was designed to prioritize diversity, making use of domain randomization to extend beyond typical real-world variations.

The procedural model's core uses two varieties of Split Grammar. The first, utilizing the CGA language, was employed to subdivide the volume to create the building mass, façades, shutters, blinds, balconies, roof, and window-bounds. From these window-bounds, a second Split Shape Grammar creates the window shapes themselves, using Bèzier spline curves to capture various geometries, such as trapezoid, arched, or circular windows. This grammar subdivides closed curves to create nested window frames and offsets them to create individual glass panes. The frame geometry is created by extruding a profile along each Bèzier. The profiles are selected from a randomly selected hierarchy of profiles. Parts of the window geometry hierarchy can be translated and rotated to 'open' windows by sliding or hinged mechanisms. This multi-grammar approach allows for detailed and varied window designs, while the generation remains algorithmically robust at scale.

The procedural model is highly parameterized, with the number of parameters ranging from 216 to 21,735 in our dataset, depending on the chosen sequence of randomly-sampled rules in the grammars. We observe that optionally reusing parameters between parts of the model can improve visual realism; for example, we see window frames sharing their material (paint or stucco) with walls, or adjacent windows having similar (but not identical) materials. In the analysis section below we vary the distributions of these parameters to ablate and experiment on our procedural model and identify the most performant factors, including materials, textures, geometry, camera position, and lighting.

Textures are primarily procedural shaders [7], controlling materials such as wood, brick, or glass applied to appropriate object classes. To add realism, we captured exterior clutter, such as building signage, vehicles, and trash cans, through a number of sources, including pre-existing datasets and LiDAR/RGB scanning (Appendix Fig. 12) with bitmap textures.

Scene lighting is supplied by a combination of skybox emission, sun-lamp, and optional interior sources. We use interior and exterior panoramic images for the skybox and interior-box to create realistic background environments for our geometry. The camera is strategically positioned in order to capture the entire window from a predominantly frontal view; we use a variety of camera distances and adjust the field of view to frame the target window in the façade. Additional details of the procedural model can be found in Appendix Section 6.

5. Estimating Procedural Model Quality

We use semantic segmentation as a benchmark to evaluate the quality of a procedural model. This method involves training segmentation models on labeled images generated from the procedural model, evaluating model performances on a holdout (*test*) set of labeled real-world images, and providing a robust measure of the synthetic data's ability to accurately reflect real-world scenarios.

For this approach to be successful, we must align our semantic labels with both real-world images and images generated by a procedural model. We established ten broad categories such as 'window-pane', 'window-frame', and 'wall frame' (see Appendix Section 7.7). This categorization mirrors the elements that our procedural model can realistically generate. Furthermore, we label each unique instance within the images, as demonstrated in Fig. 3, making the data suitable for panoptic segmentation even though we only evaluate the semantic segmentation task. This detailed set of labels extends traditional segmentation datasets, which typically provide a single 'window' label, with occasional additions like 'shutter' or 'blind' [6, 25, 36]. Our labeling allows consistent annotation across varied architectural styles but also ensures compatibility with the capabilities of procedural modeling.

When utilizing mIoU to evaluate procedural model performance, inspecting plots such as Figures 6 and 7 can be insightful. These visual analyses aid in identifying parameters that optimize the model's performance. Additionally, Spearman's rank correlation is applied to quantify the impact of changes in model parameters (such as textures and lighting) on the synthetic image quality.

6. Analysis

In all following experiments, we fine-tune a BEiT 'base' model that was pre-trained on ImageNet-1k, trained on ImageNet-21k, and fine-tuned on data from some variation of our procedural model. We evaluated the mIoU over 10 labels, excluding the 'unlabeled' category. All cropped window-images are resized to 512 pixels square; examples

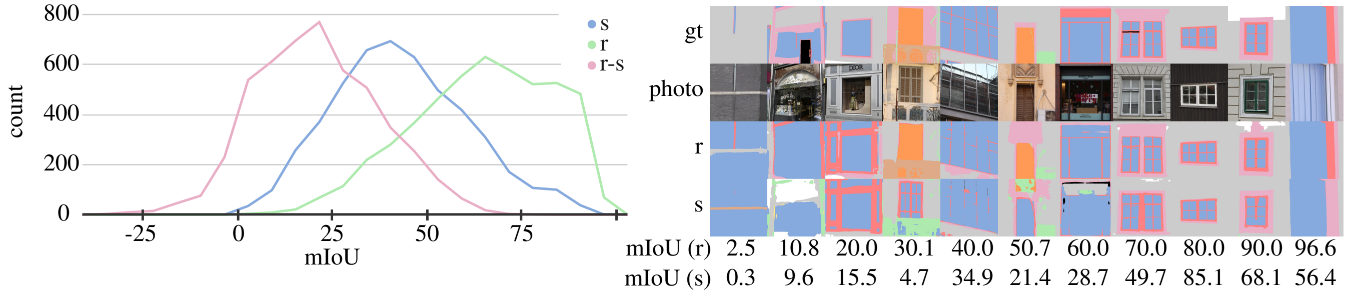

Figure 4. Left: histogram of per-image mIoUs showing the distribution of labeling results for a model trained on n = 2,048 synthetic (s) and real (r) images. We also show the difference between the model mIoU's per image (r-s). The mIoU was evaluated on 4,906 real images. Right: random samples of the labeling quality for networks trained on real and synthetic data; the first sample with an mIoU above each decile was selected.

are shown in Fig. 4.

The performance of our procedural model is best understood in the context of regional performance disparities within real-world data. We contextualize the procedural model by comparing it to subsets from various regions of real-world data in Table 2 shows. Each set consists of 1,024 training images and 300 test images. Models trained on synthetic data have a narrower performance range (29.33 to 35.02 mIoU) compared to those trained on locale-specific data (25.23 to 61.85 mIoU). Notably, synthetic data can surpass real data from a different locale, as seen in the comparison of England and Egypt results. This narrower range seems to indicate synthetic data has not over-fit any particular region. The overall mIoU gap on the global dataset underscores the procedural model's limitations. The combined training set of real-world images evaluated to 53.79 mIoU, which is an upper limit on what one should expect from any model. The synthetic data yields 31.23 mIoU on the global test set, whereas other locales varied from 37.76–51.22 mIoU for the 'other' set. Given the global set includes holdout images from each locale, it is not surprising the mIoU's are slightly higher. This table indicates that the procedural model is of comparable quality to choosing a single locale.

6.1. Procedural Model Variations

We assess the impact of synthetic dataset variations on a segmentation model by comparing against our baseline model. For each variation we render $2,048$ training examples and use the same geometry, lighting, and rendering settings as the baseline unless specified. For each variation, we render 2,048 training examples. Empirically, we find that the performance of synthetic data levels off at this number of examples (see Appendix Fig. 7) and evaluation is faster if the set is kept smaller. Evaluation takes place on a test partition of 4,906 real-world images; the mIoU of the baseline model is 32.58 and real photos have an mIoU of 58.69. To gauge each variation's importance, we either the relative mIoU range as a percentage of the baseline, or report the mIoU Spearman's rank correlation (r_s). Detailed results are in Appendix Section 7 and summarized here.

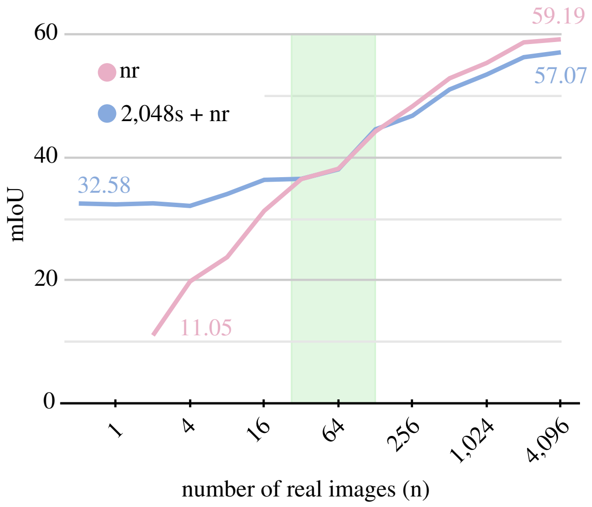

Figure 5. The effect of varying real-world dataset sizes on mIoU with (blue) and without (red) an additional 2,048 synthetic samples. The green area bounds the range in which adding real data neither harms nor improves performance; the right-most point of which has n = 152 with an mIoU of 44.96. At larger datasets, synthetic data slightly reduces mIoU relative to only using real-world data.

Rendering samples. We evaluated the impact of samples per pixel (spp) on render quality, noting diminishing returns beyond 256spp (Fig. 6). Render times scale from 6.4s at 1spp to 85.2s at 512spp. A strong correlation ($r_s = 1, n = 10$) exists between spp and mIoU, with a 68% change in mIoU scores relative to baseline, underlining the importance of spp. In this experiment no denoising was performed, however, our baseline model used 256spp and a powerful neural denoiser.

		test						
		global	**Austria**	**Egypt**	**UK**	**USA**	**other**	**synthetic**
train	**global**	53.79	58.20	56.85	42.64	50.16	54.74	31.21
	Austria	39.49	58.23	28.36	34.40	35.67	41.32	21.87
	Egypt	47.51	38.32	61.85	33.02	38.01	49.75	31.34
	UK	37.76	40.68	25.23	38.56	35.39	37.64	28.51
	USA	41.08	42.50	27.39	33.30	50.08	39.39	27.02
	other	51.22	48.54	39.47	37.44	41.33	52.74	28.39
	synthetic	31.23	29.53	29.33	32.15	34.17	35.02	62.12

Table 2. mIoU for different splits of the real labeled data on the segmentation task. Trained on 1,024, tested on 300 samples. *global* is a mixture of all the real data; *other* data is from locales outside of Austria, Egypt, UK, or USA. In this experiment our synthetic network is similarly trained on 1,024 samples from our baseline synthetic model.

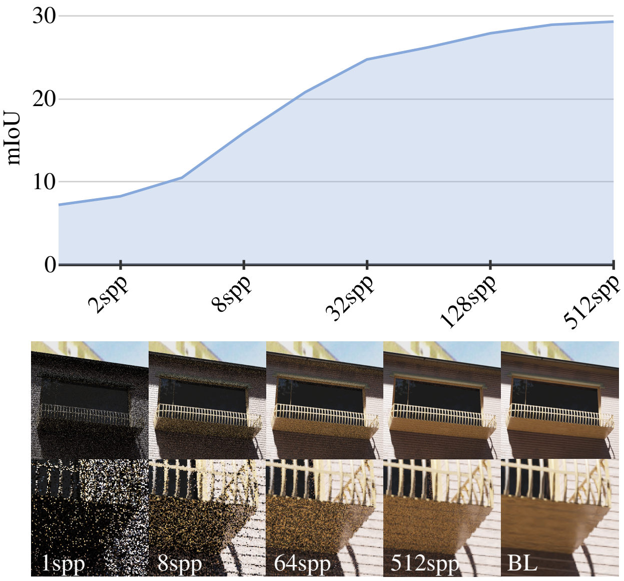

Figure 6. Top: The impact of rendering samples per pixel (spp) on segmentation task accuracy. Bottom: example renderings at different spp with zoomed section of rendering. BL = baseline.

Mixed Materials. We experimented with 9 material variations, including uniform gray, edge shaders, and random textures. Some tests used a single material for the entire scene, others assigned different materials per object. Simplifying materials, such as rendering only the albedo channel, resulted in a significant performance drop. Appendix Fig. 18 shows that any material restriction led to at least an 18% mIoU decrease from the baseline.

Lighting. We examined 8 lighting model variations and their mIoU impact, as detailed in Appendix Fig. 19. Lighting models like albedo-only were crucial, while lighting conditions such as night-time had moderate importance. Conditions varied mIoU by 15.35% from baseline. Daylight-only training decreased baseline mIoU by 1.04% relative to baseline.

Camera. We experimented with the distribution of camera positions. These 7 variations used a simple model which sampled a camera position over a circle, of radius $r = \{0...96\}$ meters, truncated at the floor plane. The circle is positioned 5 meters from the wall, directly in front of the window. The camera's field of view is adjusted to the apparent window size. We observed limited impact on mIoU as r changes; peak task performance was at $r = 12m$.

Window Geometry. We ran 7 tests with varying window dimensions and shapes, including square and non-rectangular windows. The mIoU impact was minor, fluctuating by up to 5.7% relative to the baseline (1.86 absolute), with the best variation 1.2% worse relative to the baseline (-0.04 absolute). Small features, though noticeable to humans, had a weak impact on model performance.

Labels Modeled. In developing our procedural synthetic data generator, we prioritized labels by size, starting with *wall* and ending with *open-window* (Appendix Section 4). This enabled the assessment of mIoU at nine developmental stages (Fig. 7). Adding smaller classes later showed diminishing returns and occasionally reduced single-label accuracy (Appendix Fig. 22).

6.2. Additional Experiments

In our segmentation experiments, we employed the BEiTv2 base model [1, 30] for its proficiency in generating accurate results with minimal data. However, the trends we observed are consistent across various models, including DeepLabv3+ [10]. In the Appendix Section 5, we discuss two experiments that study the impact on the segmentation task when training (and testing) with different mixes of real and synthetic data. The first experiment in Appendix Fig. 7 explores segmentation performance when training and testing on different amounts of real or synthetic data. We observe that synthetic data saturates (stops improving with additional data), while real-world data continues to benefit from additional samples. The second experiment, in Appendix Fig. 8, illustrates the impact of different mixtures of

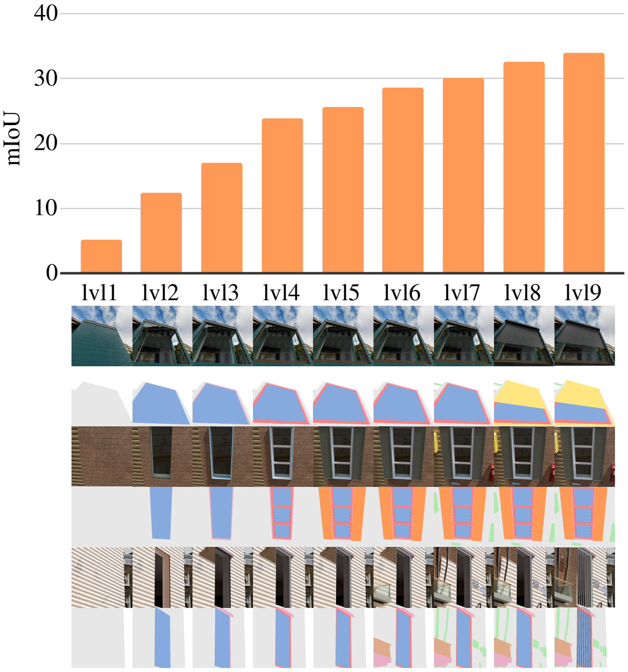

Figure 7. Top: The progressive impact of incorporating additional labels into the procedural model, with label subsets expanding from lvl1 to lvl9. The sequence begins with 'walls' only (lvl1), adding 'window panes' (lvl2), 'wall frames' (lvl3), 'window frames' (lvl4), 'shutters' (lvl5), 'balconies' (lvl6), 'miscellaneous objects' (lvl7), 'blinds' (lvl8), and 'bars' (lvl9). The full model includes additional features beyond lvl9, such as interior dressing, open windows, and windows without glass. Bottom: Examples corresponding to each incremental level of model complexity.

real and synthetic data, when tested on real data. Mixing synthetic and real data can be advantageous when there is little real data, but adding large amounts of synthetic data does not help task performance. We conclude that no volume of synthetic data can overcome this domain gap.

An exploration of different architectures and data partitions is presented in Appendix Fig. 9 and Table 2 - we train an older convolution network without pre-training (DeepLab3+ [10]), a 'large' BEiTv2 model, an architecture which uses real data labels during training to improve performance (Label Adaptation [39]), another technique which uses unlabeled data to improve performance (MIC [20]), adjusting the colors of the training dataset (Histogram Matching [17]), and creating an 'easy' real data partition. Under these approaches, the synthetic-real domain gap persists, suggesting that network architecture and data partitions are an orthogonal research direction to improving the quality of synthetic procedural models.

7. Acknowledgements

We thank our lead photographers Michaela Nömayr and Florian Rist, and engineer Prem Chedella, as well as our contributing photographers: Aleksandr Aleshkin, Angela Markoska, Artur Oliveira, Brian Benton, Chris West, Christopher Byrne, Elsayed Saafan, Florian Rist, George Iliin, Ignacio De Barrio, Jan Cuales, Kaitlyn Jackson, Kalina Mondzholovska, Kubra Ayse Guzel, Lukas Bornheim, Maria Jose Balestena, Michaela Nömayr, Mihai-Alexandru Filoneanu, Mokhtari Sid Ahmed Salim, Mussa Ubapa, Nestor Angulo Caballero, Nicklaus Suarez, Peter Fountain, Prem Chedella, Samantha Martucci, Sarabjot Singh, Scarlette Li, Serhii Malov, Simon R. A. Kelly, Stephanie Foden, Surafel Sunara, Tadiyos Shala, Susana Gomez, Vasileios Notis, Yuan Yuan, and finally labelyourdata.com for the labeling.

8. Conclusion

We have introduced a new dataset of 75,739 photos (2.09 terapixels), of which 9,002 photos are semantically labeled, as well as a high-quality procedural model that closely approximates real-world variation, making it effective for image segmentation tasks. Together these components of WinSyn have applications in unsupervised domain adaptation (using the large amount of unlabeled data), super-resolution (via the thigh resolution images), learning from RAW sources, generative modeling, 3D reconstruction or depth recovery (using the synthetic depth or 3D geometry from the procedural model), as well as learning from images containing transparent or specular materials (i.e., window panes/glass balconies are often transparent).

We systematically explored the effect of variations on our model on mIoU over 64 variations and 156,903 renders. The mIoU performance gap between our synthetic and real-world data is comparable to inter-country differences with largely different architectural styles. However, the difference between synthetic data and real data is still much larger than desired. This gap is not sufficiently reduced by either our work or other network architectures or dataset scale. We, therefore, believe that WinSyn provides a timely and efficient testbench from which researchers can iterate quickly to explore graphics contributions to the synthetic data problem. Our work contributes a sizable and versatile dataset that can be the basis of exciting and much-needed progress in the area of procedural graphics for synthetic data generation in machine learning. The key value of our dataset is that as a new benchmark of simulated and real images, it enables others to study the problem at an approachable level of complexity.

References

[1] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022. 7

[2] Björn Barz and Joachim Denzler. Wikichurches: A fine-grained dataset of architectural styles with real-world challenges. *arXiv preprint arXiv:2108.06959*, 2021. 2

[3] *Cycles: Open Source Production Rendering*. Blender Foundation, 2023. 5

[4] Gabriel J. Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV (1)*, pages 44–57, 2008. 3

[5] Clemens-Alexander Brust, Sven Sickert, Marcel Simon, Erik Rodner, and Joachim Denzler. Efficient convolutional patch networks for scene understanding. In *CVPR Workshop on Scene Understanding (CVPR-WS)*, 2015. 2, 3

[6] Clemens-Alexander Brust, Sven Sickert, Marcel Simon, Erik Rodner, and Joachim Denzler. Efficient convolutional patch networks for scene understanding. In *CVPR Workshop on Scene Understanding (CVPR-WS)*, 2015. 2, 3, 5

[7] Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *Acm Siggraph*, pages 1–7. vol. 2012, 2012. 5

[8] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*, pages 611–625. Springer, 2012. 3

[9] J Chen, R Stouffs, and F Biljecki. Hierarchical (multi-label) architectural image recognition and classification. 2021. 2

[10] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 7, 8

[11] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2023. 5

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[13] D. Dai, H. Riemenschneider, G. Schmitt, and L. Van Gool. Example-based facade texture synthesis. In *International Conference on Computer Vision (ICCV)*, 2013. 2

[14] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 3

[15] Epic Games, Inc. MetaHuman Realistic Person Creator, 2021. 2

[16] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016. 3

[17] Rafael C Gonzales and Paul Wintz. *Digital image processing*. Addison-Wesley Longman Publishing Co., Inc., 1987. 8

[18] Daniel Cabrini Hauagge and Noah Snavely. Image matching using local symmetry features. In *2012 IEEE conference on computer vision and pattern recognition*, pages 206–213. IEEE, 2012. 2

[19] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. BOP challenge 2020 on 6D object localization. *European Conference on Computer Vision Workshops (ECCVW)*, 2020. 3

[20] Lukas Hoyer, Dengxin Dai, Haoran Wang, and Luc Van Gool. Mic: Masked image consistency for context-enhanced domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11721–11732, 2023. 8

[21] Jian Kang, Marco Körner, Yuanyuan Wang, Hannes Taubenböck, and Xiao Xiang Zhu. Building instance classification using street view images. *ISPRS journal of photogrammetry and remote sensing*, 145:44–59, 2018. 2

[22] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2017. 3

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 3

[24] Roman Kaskman, Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects. *International Conference on Computer Vision (ICCV) Workshops*, 2019. 3

[25] F. Korč and W. Förstner. eTRIMS Image Database for interpreting images of man-made scenes. Technical Report TR-IGG-P-2009-01, 2009. 2, 5

[26] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 2

[27] Jose Llamas, Pedro M. Lerones, Roberto Medina, Eduardo Zalama, and Jaime Gómez-García-Bermejo. Classification of architectural heritage images using deep learning techniques. *Applied Sciences*, 7(10):992, 2017. 2

[28] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[29] Pascal Mueller, Peter Wonka, Simon Haegler, Andreas Ulmer, and Luc Van Gool. Procedural modeling of buildings. *ACM Trans. Gr.*, 25(3):614–623, 2006. 5

[30] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers, 2022. 7

[31] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies

and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 2

[32] Alexander Raistrick, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, et al. Infinite photorealistic worlds using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12630–12641, 2023. 3

[33] Hayko Riemenschneider, Ulrich Krispel, Wolfgang Thaller, Michael Donoser, Sven Havemann, Dieter Fellner, and Horst Bischof. Irregular lattices for complex shape grammar facade parsing. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1640–1647. IEEE, 2012. 2

[34] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 3

[35] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 2

[36] Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *Proc. GCPR*, Saarbrucken, Germany, 2013. 3, 5

[37] Stephen Tyree, Jonathan Tremblay, Thang To, Jia Cheng, Terry Mosier, Jeffrey Smith, and Stan Birchfield. 6-dof pose estimation of household objects for robotic manipulation: An accessible dataset and benchmark. In *International Conference on Intelligent Robots and Systems (IROS)*, 2022. 3

[38] Peter Wonka, Michael Wimmer, François Sillion, and William Ribarsky. Instant architecture. *ACM Trans. Gr.*, 22 (3):669–677, 2003. 5

[39] Erroll Wood, Tadas Baltrusaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J. Cashman, and Jamie Shotton. Fake it till you make it: face analysis in the wild using synthetic data alone. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3661–3671. IEEE, 2021. 2, 3, 8

[40] Zhe Xu, Dacheng Tao, Ya Zhang, Junjie Wu, and Ah Chung Tsoi. Architectural style classification using multinomial latent logistic regression. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 600–615. Springer, 2014. 2

[41] K. Zhao, Y. Liu, S. Hao, S. Lu, H. Liu, and L. Zhou. Bounding boxes are all we need: Street view image classification via context encoding of detected buildings. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–17, 2021. 2

[42] Peihao Zhu, Wamiq Reyaz Para, Anna Frühstück, John Femiani, and Peter Wonka. Large-scale architectural asset extraction from panoramic imagery. *IEEE Transactions on Visualization and Computer Graphics*, 28(2):1301–1316, 2022. 2