Abstract

Creating building energy models of neighborhoods and cities is becoming increasingly important as cities seek to use simulation to inform their decarbonization strategies for their building stock. To facilitate urban energy modeling, we introduce a scalable method to capture geometric properties, including existing buildings’ window-to-wall (WWR) ratios, following five main steps: (1) Use automated drone flight planning and aerial image capture to rapidly collect aerial imagery of neighborhoods or blocks. (2) Use photogrammetry to extract textured 3D models. (3) Isolate buildings and extract texture maps. (4) Use a custom-trained Machine Learning (ML) model to segment texture maps and identify windows. (5) Extract window and envelope areas for Building Energy Modeling. We report a mean absolute error for WWR predictions of our ML model of 8.99% compared to manually labeled images.

Highlights

- Discusses the use of photogrammetry for urban building energy models.
- Trains a custom image segmentation model to identify windows on textured 3D models.
- Detect exact window geometry and position on buildings.
- Scalable method to fill data gaps in urban building energy models setup.

Introduction

The built environment is responsible for 37% of global greenhouse gas emissions annually (UN Environment Programme, 2021), and its decarbonization is an urgent, massive challenge requiring immediate attention to limit the adverse effects of climate change (OECD, 2012). Reducing building sector emissions will largely depend on addressing the energy consumption of already existing buildings. Urban building energy models (UBEMs) can help modelers understand energy use in the existing building stock and inform energy policy and large-scale retrofitting. However, creating UBEMs remains difficult as available data often lack essential geometric information, such as window-to-wall ratios. To capture this information, Cao et al. (2017) introduced a method that generates facade geometry from low-resolution single-view aerial photographs; others extracted the window geometry profile projection method (Sung Chun Lee & Nevatia, 2004) or reconstruct façades by detecting symmetry (Xiao et al., 2008). Further, laser scanning has been introduced in building geometry modeling (Nan et al., 2010, Zheng et al., 2010 O’Donnell et al., 2019), as well as a semi-automatic method that fuses laser scan data with photographs (Li et al., 2011).

This paper introduces a novel and scalable method that uses the combination of conditional General Adversarial Networks (cGANs) and Segment Anything Model (SAM) for texture map synthesis (Chaudhuri et al., 2021) of photogrammetry models to segment the texture map of building envelope into façade, window, and roof geometry for UBEMs. Compared to previous research that investigated WWR reconstruction by capturing and rectifying images for single façades, leveraging unrolled texture maps of full building envelopes for image segmentation is advantageous as it captures the entirety of a building envelope in a single image and is, therefore, easier to scale and automate. Previous WWR reconstruction work mostly relied on methods from signal processing (Cao et al., 2017) and rely on the assumption that windows are rectangular and appear in a somewhat regular pattern on the façade. This has limitations with more complex building shapes. Hence, we propose leveraging recent advances in machine learning and computer vision in conjunction with the previously described unrolled photogrammetry texture maps. The GAN training in our method uses pix2pix (Isola et al., 2017), a type of GAN used for image-to-image translation tasks. It has a generator that produces output images similar to the target domain and a discriminator that distinguishes between real and generated images. During training, the generator and discriminator play a game where the generator tries to produce images that fool the discriminator, and the discriminator tries to correctly identify the generated images as fake. This process continues until the generator can produce high-quality images indistinguishable from the target domain. To enhance the clarity of identified window geometries, we incorporate the Segment Anything Model (SAM) (Kirillov et al., 2023). SAM is pre-trained with a large dataset that enables powerful zero-shot generalization without the need for additional training. SAM consists of an image encoder and a prompt encoder. The two information sources are later combined in a mask decoder that eventually predicts the segmentation masks.

Our proposed method follows five main steps: (1) Use drone and automated flight planning to collect aerial imagery of neighborhoods or blocks rapidly. (2) Use
Photogrammetry to extract textured 3D models. (3) Isolate buildings and extract texture maps. (4) Use a custom-trained ML model to segment texture maps for window identification. (5) Extract window and envelope areas for BEM.

Methods
Our method involves capturing areal photographs that are processed with photogrammetry workflows to generate textured 3D models of buildings. We isolate 3D models of buildings and extract their texture maps, which include information about the entire envelope. Next, We apply a custom-trained image segmentation model to texture maps to identify windows, opaque façade elements, and other parts of the building envelope. Unlike other approaches that operate on rectified images of every individual façade, our approach isolates and compiles all information of the entire envelope in only one texture map, resulting in improved data processing efficiency. This method can also capture façades that may be inaccessible from ground level with no perspectival distortion. Our image segmentation model is based on the combination of pix2pix and SAM and has been trained on 45 labeled image pairs. The following sections describe each step in more detail.

Aerial Building Image Capture using Drones
Image collection for photogrammetry must follow a few simple guidelines so that a 3D model of the surveyed area can be created successfully. Images must be captured with sufficient overlap (60-80%) (Agisoft. 2022) so that common points can be found by the photogrammetry software. For isolated objects, like a single building, capturing photos on a cylindrical path around the object covering all possible view angles is recommended. In order to capture more extensive areas, we used a DJI Mavic 2 (DJI, 2018) drone programmed with the Pix4Dcapture (Pix4D, 2021) mobile application to automatically survey a larger area with a double-grid flight path with an image-capturing frequency of 70%, and camera angle with 55% (Figure 1). To cover an area of 10000 square meters (100x100m), this typically yields 300 images. The drone used captured images at 5472 × 3648 pixels with 72 ppi and geolocation data from the drone’s GPS sensor.

Photogrammetry is a technique that uses photographs to create 3D models of objects or environments. It works by analyzing the overlapping areas of multiple photos of the same subject and using the differences in perspective to triangulate the position of each point in 3D space. The resulting 3D point cloud can then be used to generate a 3D model or map. In our methodology, we utilize Agisoft Metashape (Agisoft. 2022) software that automates most steps and produces a georeferenced 3D mesh with textures. The success rate and quality of the 3D models vary based on the images and the complexity of the object to reconstruct. The most commonly encountered issues are related to vegetation partially covering the building. Further, sharp shadows, sunlit shiny surfaces, or leafless vegetation can confuse photogrammetry algorithms leading to flawed camera alignment and therefore degraded model quality. In this study, it was not entirely possible to avoid these issues, given the large extent of the area that had to be covered. We, therefore, verified model accuracy by using a Leica BLK360 laser scanner to capture high-resolution LiDAR point clouds of our building façades (Figure 2) to discuss the geometric accuracy of our photogrammetry models and the feasibility of the overall modeling approach.

Building Isolation and Texture Map Optimization
The photogrammetry models obtained from Agisoft Metashape are colored with information captured in the original image series. Meshes are 3D models composed of interconnected vertices, edges, and faces that define the shape of an object. The color information is stored in the form of texture maps, 2D images from which color information is extracted using 2D coordinates assigned to each vertex of a 3D mesh, which defines how the texture is mapped onto the surface of the mesh. To make the texture maps easier to interpret for humans, it is helpful to isolate the buildings into separate models and recompute and optimize the texture maps using Smart UV Project in Blender (Blender Foundation, 2023).

To remove surrounding structures and protruding vegetation from the texture maps, we isolate buildings by inflating the geolocated footprint by a margin (the distance depends on the surrounding condition, in the range of 0m - 2m) and then cropping away all geometry.
outside the inflated footprint. Vegetation attached to the building is kept in the mesh along with other façade elements and should not impact the WWR calculation as long as the windows are not obscured. Further, the original triangular mesh is quadrangulated to optimize the mesh topology. Specifically, in most buildings, the windows are placed orthogonally on the façade surfaces; quadrangulating the mesh would allow the mesh surface to align better with the window shape. This approach facilitates subsequent mesh surface separation into windows and non-windows while retaining the accurate geometry of the original model (Figure 3).

Then the texture map is optimized with Blender’s “Smart UV Project” which divides the mesh based on angle thresholds, then recomputes the texture maps by projecting the 3D surface onto a 2D UV map. This involves automatically defining seams on the 3D surface, grouping, orienting, and aligning the UV map with the 3D model, and calculating the mapping between the 3D surface and the UV map. The resulting UV and texture map look similar to an unrolled surface and therefore are significantly easier to label for machine learning model training.

![Figure 3: Original texture map and mesh(A) and their optimized version(B) after quad remeshing.](image)

**Machine Learning for Window Detection**

To enhance the identification of windows and their edges within a texture map, we introduce a novel procedure that combines the pix2pix GAN and SAM. This hybrid pipeline leverages the strengths of both cGANs and SAM to enhance the accuracy and efficiency of window detection. We first employ the pix2pix framework to make a coarse estimate to identify the windows. To refine the segmentation results and reduce noise and blur at the window boundaries, we subsequently prompt SAM with the bounding boxes of the identified windows by pix2pix. SAM, which has been trained on a large dataset and is able to generate reliable and clean boundaries, enhances the accuracy of the segmentation process.

**Pix2pix Training Data Preparation**

We train pix2pix using our own datasets, consisting of a pair of extracted building texture maps as input and corresponding labeled images as the expected output (ground truth). The trained model learns the intricate image structures and patterns to detect the windows in the texture map automatically.

Our training dataset is composed of 45 building texture maps and labeled images. Ground truth images are manually labeled, segmenting the collected building texture maps into pixels representing windows, window edges, and other elements – each with a specific color label. For the scope of this paper, we focus solely on windows, in which we adopted a labeling scheme that uses three colors to represent the window, window edge, and other components (Figure 4). This approach significantly reduces the errors associated with the deep learning model operating on a relatively small and preliminary dataset by focusing on windows that are easily distinguishable with some structural differences in the image. Furthermore, this approach has also significantly increased efficiency in manual labeling. To further segment the building geometry into roof's and facades, we introduce a vertex normal-based grouping explained in the following section.

![Figure 4: Data set preparation for façade element detection workflow and color pattern for image labeling](image)

**Data Augmentation**

In this section, we address the challenge of a limited original dataset due to the laborious labeling of texture maps. To overcome this limitation, we transform our sparse training data using OpenCV. The proposed approach transforms each image pair (input and expected output) through a randomized scaling range of 1.0 to 3.0 and rotation range of 0 to 10 degrees. This transformation process generates 50 distinct variations for each image pair, therefore increasing the dataset from 45 to 2250 texture map pairs, which significantly improves the performance of the trained model. Further, the image variation process results in greater diversity in our original dataset. By providing these additional images, we aim to improve the robustness and accuracy of our machine-learning model. This can be crucial for the model's performance in real-world applications where windows can vary greatly in size and angle. After this data augmentation process, the model is trained using a dataset of square image pairs of 512px x 512px resolution based on the pix2pix GAN implementation. Higher-resolution versions of pix2pix exist and may be used in future research.

**cGANS Segmentation**

In order to evaluate the generalizability of cGANS, we conducted experiments on diverse datasets encompassing residential, commercial, and public buildings. The training dataset resolution is limited to 512 x 512 pixels. During the data augmentation, a zoom scale of 1.0 – 3.0...
was applied to the original image to prevent insufficient resolution when identifying relatively small windows in the overall texture map. We crop the texture maps using 3×3 grids (each of 512×512 pixels) and apply the trained model to segment the image tiles. Subsequently, the segmented tiles were combined back to generate the complete segmented texture map. The tiles preserve sufficient resolution for the model to identify and segment window details and approximately match the average cropping of the image variation generation. A finer cropping could potentially split complete façade textures in the original map into unidentifiable pieces.

**SAM Optimization**

The trained pix2pix model learns the intricate image structures and patterns to automatically detect the windows in the texture map. While the trained GAN model generally demonstrates that it can identify the window locations within the texture maps, there are shortcomings. Due to the limited training data that lacks cases and variety, our GAN cannot always accurately define clear edges and thus generates blurry boundaries between windows and opaque facades. Given that capturing additional training data was not feasible within the scope of this publication, we seek to overcome this limitation by prompting SAM to optimize the segmentation using the bounding boxes of windows identified by the pix2pix model. Initially, we extract the windows identified by the pix2pix model and generate corresponding bounding boxes for each window. We implement a filtering process to improve the prompts’ quality and reduce noise. Bounding boxes that are very small, likely representing noisy results from pix2pix, are discarded. Additionally, we group together bounding boxes that are in extreme proximity to each other, of which pix2pix fails to identify as one connected window due to noise present in the texture map. (Figure 5). Subsequently, the resulting bounding boxes are prompted to SAM, which generates masks representing the segmented regions corresponding to the identified windows. To obtain the final segmentation results, we utilize OpenCV to apply colorization to the masks. (Figure 5).

**Window-to-Wall Ratio (WWR) Calculation**

Following the semantic segmentation of the texture map, the number and size of the windows on the building envelope can be estimated. The simplest approach would be to count the pixel of a specific color in the texture map directly. In our approach, however, we apply the segmented texture map back to the building model, divide up the envelope into façade segments and roof surfaces and then calculate the window areas and opening ratios for each surface. This allows us to preserve the correct opening ratios on each building energy model surface.

**Window Surface Separation**

Our objective is to separate the mesh surfaces identified as windows from the original model for area calculation, which provides further potential for reconstructing the mesh into a clean energy model with the window mapped at the exact location. The proposed approach extracts the color of the center point of a mesh surface being identified as a window and traverses through all mesh surfaces to find and group surfaces with the same color. Since we have quadrifid the mesh beforehand, the resulting boundaries of the grouped surfaces are orthogonal. This enables us to calculate the window areas later and precisely determine window dimensions.

**Window Area and WWR Calculation**

(Figure 5: Comparison of pix2pix predicted result and optimized SAM predicted result.)
We leverage the structure semantics of buildings and utilize the surface normals to group facades by orientation and separate them from roof and ground surfaces.

We identify surfaces by establishing a threshold of 75 degrees angle for the angle between the normal vector and the ground. Surfaces with normals exceeding 75 degrees would be considered facades.

Furthermore, we apply the same method to the window surfaces, excluding those not facing sideways when calculating the window area. After the wall and window area calculation, we then compute the overall WWR of a specific building.

**Results**

To evaluate the validity of the proposed WWR detection methods, we collected and compared three sets of WWR measurements on eight buildings, using (1) surface normal calculation based on a manually labeled texture map, (2) the proposed method using pix2pix—surface normal calculation based on predicted image segmentation and (3) the proposed method using pix2pix with SAM (Table 1). Then, the error rate is computed based on the comparison between (1) and (2), and (1) and (3). The eight buildings represent different building typologies with various shape complexity, from single-family residential houses to mid-rise offices. We also qualitatively evaluate the result through visual comparison (Figure 6).

**Table 1: WWR validation based on different models.**

<table>
<thead>
<tr>
<th>WWR Manual Label (%)</th>
<th>WWR pix2pix (%)</th>
<th>Error Rate 1 (%)</th>
<th>WWR SAM Optimized (%)</th>
<th>Error Rate 2 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.32</td>
<td>-15.26</td>
<td>11.07</td>
<td>-10.15</td>
</tr>
<tr>
<td>2</td>
<td>6.87</td>
<td>-15.43</td>
<td>7.57</td>
<td>10.19</td>
</tr>
<tr>
<td>3</td>
<td>4.99</td>
<td>3.81</td>
<td>5.07</td>
<td>1.60</td>
</tr>
<tr>
<td>4</td>
<td>4.38</td>
<td>-8.22</td>
<td>4.18</td>
<td>-4.57</td>
</tr>
<tr>
<td>5</td>
<td>17.85</td>
<td>-30.53</td>
<td>15.33</td>
<td>-14.12</td>
</tr>
<tr>
<td>6</td>
<td>7.68</td>
<td>-21.09</td>
<td>6.87</td>
<td>-10.55</td>
</tr>
<tr>
<td>7</td>
<td>8.37</td>
<td>-8.36</td>
<td>8.65</td>
<td>3.35</td>
</tr>
<tr>
<td>8</td>
<td>8.05</td>
<td>-13.04</td>
<td>6.65</td>
<td>-17.39</td>
</tr>
<tr>
<td>A</td>
<td>-</td>
<td>-</td>
<td>14.47</td>
<td>8.99</td>
</tr>
</tbody>
</table>

As summarized in Table 1, we found the mean absolute error (A) for the pix2pix predicted image to be 14.47% and SAM optimized image to be 8.99%.

**Discussion**

This paper introduces a semi-automated, scalable, and robust method for the analysis of façade WWRs. The proposed method successfully constructs 3D building models on an urban scale and segments texture maps into the windows and façade components to analyze WWRs reliably with a mean absolute error rate of 8.99%.

In contrast to earlier techniques that relied on aerial view imagery or ground-level photographs, the use of photogrammetry models provides several advantages: Specifically, photogrammetry models enable modelers to capture all facades of a structure, including those that may be inaccessible from ground level. Moreover, photogrammetry models can help avoid potential issues related to distortion resulting from a perspective in aerial images, which may be encountered with other methods.

Our approach has several limitations and can be improved in future work. The automatic window identification may fail when too many views are blocked near or on the façade. For example, there are two common scenarios related to vegetation that may cause detection issues: (1) Vegetation with a high leaf density might block a large area of the building façade, and thus no accurate 3D model of the entire building envelope can be created. This could be solved by fusing aerial images with ground-based scans that can see behind the obstructing vegetation; (2) In winter, deciduous vegetation might be recognized as a projected texture on building facades, which creates blurry artifacts (See Figure 7) on the texture maps and disturbs the window identification.

Furthermore, complex façade decoration, distortion, and highly reflective façade materials will reduce the accuracy of model generation and result in inaccurate geometry or poorly mapped texture. Another potential for inaccuracy in the photogrammetry model are high-contrast shadows on building facades. These shadows can sometimes be dark enough to mask the underlying facades. Although this limitation could be improved by choosing overcast days for scanning, some inevitable shadows might still influence the window identification process. Another way to mitigate this issue would be to employ a drone camera with a better dynamic range.

![Figure 7: Challenges in façade element identification](image)

In the initial method, the image labeling approach involves identifying more segments, including roofs, façades, windows, and other elements. However, the increased semantic complexity of the images presented a challenge in distinguishing different façade components with the limited training dataset used in this study. Specifically, roofs and facades without windows, which lack structural difference, have proved challenging to distinguish under many situations (Figure 7), making it also difficult for human labeling. With a larger training dataset, the model could potentially segment non-window textures into walls, roofs, skylights, HVAC systems, and other façade components.

The integration of pix2pix and SAM yields substantial improvements in image segmentation quality, demonstrating promising results in window identification of the texture map. Our proposed approach presents a simple, novel, and effective method for achieving reliable...
and valid semantic image segmentation, even with a limited training dataset. However, the proposed methods may occasionally overlook certain windows that are not identified by pix2pix. Inevitably, machine learning models improve with larger training data sets. With our preliminary training data, we expect some inaccuracies to occur with semantic segmentation. However, in real-life applications, this limitation can be effectively addressed through a visual review and occasional manual touch-up of the generated texture maps performed prior to feeding the data into SAM. Moreover, the developed segmentation pipeline could facilitate semi-automated image labeling, allowing for more efficient data collection and labeling in future training.

To better facilitate future training, in addition to the proposed data augmentation methods, it is worth considering the incorporation of additional techniques such as color variation and contrast adjustment to further enhance the variety and diversity of the datasets. By introducing these additional factors, the trained model can become more robust and better equipped to handle variations in color and contrast within the texture map.

With the fast development of machine learning and computer vision capabilities, we believe the proposed approach has significant potential to facilitate large-scale window identification for urban energy modeling tasks ahead.

Conclusion

Producing the necessary input data for urban scale energy simulation purposes is a time-consuming and laborious task, particularly when it involves obtaining and reconstructing building geometry information. This paper outlines a workflow to improve the effectiveness and scalability of creating 3D models of buildings and capturing their window-to-wall ratios using a semi-automatic approach using areal image surveying, photogrammetry, and LiDAR point clouds. This enables efficient data capturing on an urban scale and provides highly detailed information for building elements. Furthermore, we transform the photogrammetry model’s texture map into a rectified and human-legible form. With the combination of pix2pix and SAM, the window outline can be identified and separated from other components on the texture map with an acceptable mean absolute error rate of 8.99%. The workflow enables the rapid creation of 3D models with window information that is needed for BEM simulation models. These photogrammetry models with LiDAR point cloud verified dimensions could potentially be converted into commonly used BEM formats such as the City Geography Markup Language (CityGML) developed by Gröger et al. (2012).

Future work in this area includes gathering larger training data sets, exploration of different image segmentation algorithms to optimize the accuracy of automatic prediction, the extension of this technique to analyze more complex building elements such as roofs, façade material, free-form surfaces, shading elements, and the application of this methods to fast larger-scale data collection. The transformation of point clouds to BIM and GIS-based formats offers the potential for simulation at district and city scales.

References


