A MACHINE LEARNING APPROACH TO ESTIMATE WINDOWS-TO-WALL RATIO USING DRONE IMAGERY

Samir Touzani[1,*], *Marc Wudunn*[2,*], *Samuel Fernandes* [1,*], *Avideh Zakhor*[2], *Rohullah Najibi*[1,2] *and Jessica Granderson*[1]

[1] Lawrence Berkeley National Laboratory

[2] University of California, Berkeley

[*] These authors contributed equally to this work

ABSTRACT

A building's window-to-wall ratio (WWR) has critical influence on heat loss, solar gain, and daylighting levels, with implications for visual and thermal comfort as well as energy performance. However, in contrast to characteristics such as floor area, existing building WWRs are rarely available. In this work we present a machine learning based approach to parse windows from drone images and estimate the WWR. Our approach is based on firstly extracting the building 3D geometry from drone images, secondly performing semantic segmentation to detect windows and finally computing the WWR. Experiments show that our approach is effective in estimating WWR from drone images.

Index Terms— Deep learning, 3D geometry, semantic segmentation, photogrammetry

1. INTRODUCTION

A building's window-to-wall ratio (WWR) has critical influence on heat loss, solar gain, and daylighting levels, with implications for visual and thermal comfort as well as energy performance. It is found that in different climates and building orientations there are optimal WWRs that result in improved operational performance. However, in contrast to characteristics such as floor area, existing building WWRs are rarely available. Estimating WWR from drone images for energy efficiency audits or other applications is a beneficial yet challenging task of which the first step is parsing the windows from the building facade. Building facade extraction from remote imagery such as drones requires processing a variety of images due to the variation of facades across different environments, the changing illumination, visual perspective, the presence of shading devices and other occlusions.

Our approach first uses photogrammetry to detect features that are shared across images and uses them in conjunction with GPS data to determine 3D points. These 3D points are projected into a 2D grid with grid cells corresponding to the building facades having a very high point density. Secondly, we train our own deep learning semantic segmentation model to detect windows on 2D images. Finally, we take the facade corner points of the extracted 3D building model, and project them onto the RGB drone camera image that corresponds to the input image with the detected windows. Once we have the 3D coordinates for the window points, we compute the WWR.

This paper is organized as follows. In Section 2 we describe previous work on the topic of parsing windows from facades and estimation of WWR. Section 3 describes our machine learning based approach and Section 4 describes experimental results from our approach. In Section 5 we conclude with some recommendations for future work.

2. PREVIOUS WORK

Building facades extraction has been studied and various methods that mostly operate on a per-pixel or super-pixel level have been proposed in computer vision [1][2]. Other methods [3], [4], [5] assume that building facades have an appropriate shape grammar. This poses strong prior knowledge on the facade of a building, but if the prior does not apply, the methods fail. Recently, deep learning has shown its power in various computer vision tasks, including image segmentation [6][7] and has outperformed traditional vision approaches in a lot of benchmarks. Schmitz et al. [8] use deep learning and treat facade parsing as a general image segmentation problem. In [9], the authors introduce laser scanning and a slicing methodology for extracting overall facade and window boundary points. Lee et al. [10],[11] combined information of line segments and calibrated facade to detect and reconstruct the windows in 3D coordinates. In [12], authors describe an approach to extract windows by analyzing geometrical characteristics of building surface. In contrast to these approaches, our approach is based on firstly extracting the building 3D-geometry from drone images, secondly using a deep learning model to perform semantic segmentation to detect windows and finally computing the WWR.

Correspondence: sgfernandes@lbl.gov

This work was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Building Technologies Office, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

3. APPROACH

3.1. Building 3D geometry extraction

In this section we will briefly describe the approach introduced in [13] and which is used in this work to extract building 3D geometry from drone images. it is based on four steps described bellow:

- *Photogrammetry:* The set of overlapping 2D RGB images captured by the drone system are processed using a photogrammetry software (i.e., Pix4D), which generates a set of data points in 3D space. This is achieved by detecting features that are shared across images and using them in conjunction with GPS data to determine 3D points. This set of data points in a 3D space is called a point cloud. Pix4D uses a fully automated process to achieve accurate 3D reconstruction based on the 2D image sets.

- *Projection of the 3D point clouds into a 2D space:* After the point cloud is extracted, the points in the cloud are projected into a 2D grid (top-down view), with a resolution of 0.1 meter (m). This permits a count of the number of points in a 0.1m x 0.1m area. As result, within this grid, the grid cells corresponding to the building facades will have a very high point density.

- *Line Detection and polygonization:* The Hough Transform is applied to the resulting 2D grid. This is a well-known method for detecting lines in an image, by converting an image to "Hough space". The extracted lines are further refined by applying an algorithm that processes the lines and extracts the segments that likely correspond to walls. Once the line segments have been extracted, polygons are constructed. As the line extraction process has various limitations and likely either misses a wall or does not extract the full line segment corresponding to a wall, an algorithm is applied to complete the polygons by taking a set of edges that would nearly form a polygon, and filling in the gaps.

- *Merging polygons and height estimation:* Once all the polygons have been completed, the adjacent polygons are recursively merged if they are close together in (average) height. To compute the average height of the polygon, we compute the average height of the points (from the 3D point cloud) in each cell, and then average the heights of the cells. This essentially "spreads out" the height computation across the full area of the polygon, minimizing the effects of objects that may lie on top of the roof.

3.2. Windows semantic segmentation

3.2.1. Dataset Creation

Few researchers have investigated image semantic segmentation of Unmanned Aerial Vehicle (UAV) or drone views of building facades and thus public data sets of sufficient size and diversity are lacking. We created our own unique dataset of images with windows labeled to perform semantic segmentation. This was important because UAV views of buildings are very specific, as they are taken from different heights and at different angles. The dataset comes from a variety of different sources including the ECP dataset [14], eTRIMS dataset [15] ISPRS dataset [16], as well as openly available drone images. In total we gathered 290 RGB images of buildings facades (i.e., street views and UAV views). ECP and eTRIMS had windows labeled on the images, while the images from ISPRS dataset and the drone openly available images did not have any labels. We used supervisely[1] which is a web tool to label images. Some images had a very large resolution (1280 by 960), so we cropped them into smaller images of building facades and resized them into 512 by 512 pixels (which is constrained by the considered neural network architecture and the GPU memory). We divided this dataset into 250 images for training the model and 40 images for validating our model. Note that the images in the validation process are from buildings that are not included in the training data set (in order to avoid a biased estimation of the accuracy).

To reduce over-fitting and to expand the number of images that cold be used for training, we employed a data-augmentation strategy. Augmentation of data is a practice in the deep learning community to create larger data sets for model training, and it can often improve the accuracy of the model. These new images are created by slightly changing the original image. For instance, making a new image a little brighter; cropping the original image; make a new image by horizontally flipping it, etc.

3.2.2. Model Architecture

In this study we employed DeepLabv3+ model [7] as a deep neural network semantic segmentation approach. This a state-of-the-art method that uses an encoder-decoder network architecture with atrous spatial pyramid pooling module to extract multi-scale contextual information by pooling features at various resolution. In the encoder-decoder structure, the encoder module extracts abstract features from the input images by gradually reducing the feature maps. The decoder module is responsible for recovering spatial resolution and location information by gradually up-sampling the feature maps. In this work, the considered output stride was equal to 16, which as was shown [17] to be the best trade-off between computational speed and accuracy. The output stride is the ratio of input image spatial resolution to the final output of the encoder. In the decoder module the encoder features are first bilinearly up-sampled by a factor of 4 and then concatenated with the corresponding low-level features maps of the same resolution from the encoder module. After the concatenation a few 3 by 3 convolutions are applied to refine the features followed by a bilinear up-sampling by a factor of 4. In this work,

[1] https://supervise.ly/

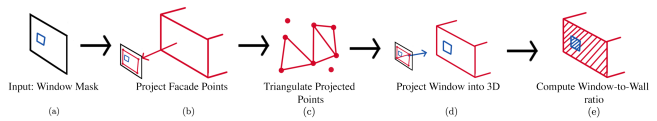


Fig. 1: Flowchart for the WWR extraction process.

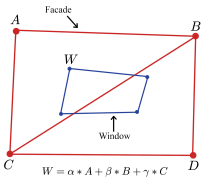


Fig. 2: Triangulation of the projected facade points with the facade in red and window in blue.

a pretrained (on ImageNet dataset [18]) ResNet-101 architecture [19] was used as DeepLabv3+ network backbone. The Adam (adaptive moment estimation) optimization algorithm [20] was used with a starting learning rate set to 0.0001, the exponential decay rate of the first moment was set to 0.9 and the second moment to 0.999. The learning rate was decayed every 25 epochs by a factor of two. The batch size was set to 16 and the number of epochs to 100.

To train the DeepLabv3+ network, a combination of two loss functions were used, the cross-entropy loss function and the Dice loss function. The cross-entropy loss is defined as:

$$L_{CE} = -\sum_{x_i \in X} log(p(x_i)) \tag{1}$$

where X is the training sample, and and $p(x_i)$ is the pixel-wise soft-max over the last DeepLabv3+ layer. The Dice loss function for multiclass segmentation, also known as the generalized Dice loss [5], is defined as:

$$L_{Dice} = 1 - 2\frac{\sum_{c \in C}\sum_{x_i \in X} p_c(x_i) r_c(x_i)}{\sum_{c \in C}\sum_{x_i \in X}(p_c(x_i) + r_c(x_i)) + \epsilon} \tag{2}$$

where ϵ is a small value added for numerical stability (set to 10^{-6}), C is the number of classes, r_c is equal to 1 if the pixel corresponds to class c and equal to 0 otherwise, p_c is the soft-max prediction for class c. Therefore, the loss function used for training models in this work is defined as:

$$L_{dicece} = w_{CE} L_{CE} + w_{Dice} L_{Dice} \tag{3}$$

where w_{CE} and w_{Dice} are the weights to each component of the loss function, and are both set to be 0.5 in this work.

3.3. Window-to-Wall Ratio Estimation

The extracted building 3D geometry and the detected windows are used to extract the window-to-wall ratio (WWR). A block diagram for the entire process is shown in Figure 1. For this approach, we assume that windows have already been detected on a 2D image as depicted in Figure 1(a), and we simply wish to compute their areas in 3D along with the area of their corresponding facades. To do so, we take the facade corner points of the extracted 3D building model, and project them onto the RGB drone camera image that corresponds to the input image with the detected windows. This is shown in Figure 1(b). The projective distance for each of the corners of the facade is stored. This will be used to determine the ordering of the facades along the optical axis. An example for the projection on one RGB image is shown in Figure 3(b).

We use a mask for the windows in a 2D image as input. Then, for each facade projected and a corresponding image, we determine whether the facade contains a window by checking whether the window points lies within the projected facade. Since we are not explicitly determining whether a facade is occluding another, a window may lie on many of the projected facades, so we need choose the closest such facade to the optical center to find the facade the windows actually lie on. This is done by choosing the facade with the minimum projective distance for the particular RGB image. Once we have determined the correspondence between the facades and their respective windows in 2D, we wish to use this information to back-project the windows into 3D and onto the corresponding 3D facade. Since the facades could have an arbitrary shape, especially the roof polygons, we perform a triangulation of all the projected facade corners via Delaunay Triangulation in 2D, as shown in Figure 1(c). We then determine the resulting triangles that each of the window vertices lie within. This allows us to obtain the position of each window point in barycentric coordinates, i.e. relative to the facade corners, as depicted in Figure 2 where the example window point W can be written in terms of the facade points A, B and C with the equation:

$$W = \alpha A + \beta B + \gamma C \tag{4}$$

For $\alpha + \beta + \gamma = 1$. Then, using the barycentric coordinates of the window points, we can compute a position for the window points in 3D space, as shown in Figure 1(d). This is done by plugging in the coordinates of the 3D facade points from the 3D building model into the same barycentric equation, i.e. replacing the 2D facade points A, B, C with the corresponding 3D facade points, and using the same α, β, and γ. Once we have the 3D coordinates for the window points we can compute the window to wall ratio by simply calculating the surface area of the facade using its 3D coordinates, along with the surface area of each of its windows with their 3D coordinates. This is shown in Figure 1(e).

4. EXPERIMENT

Deep semantic segmentation accuracy

Several DeepLabv3+ models were trained using several data augmentation strategies. The most accurate was the one that used both the pixel and the spatial augmentations (i.e., horizontal flip). The accuracy of the selected model was evaluated on the validation sample using F1 score and was equal to 0.81.

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (5)$$

where TP, FN and FP are respectively the true positive, the false negative and the false positive.

Experiment dataset

We used an openly available dataset provided by Pix4D [2]. This data set is a collection of 85 drone images from an office building collected following a circular flight path. The ground truth windows surfaces where measured using the 3D model generated by photogrammetry and the measurement tools provided by Pix4D.

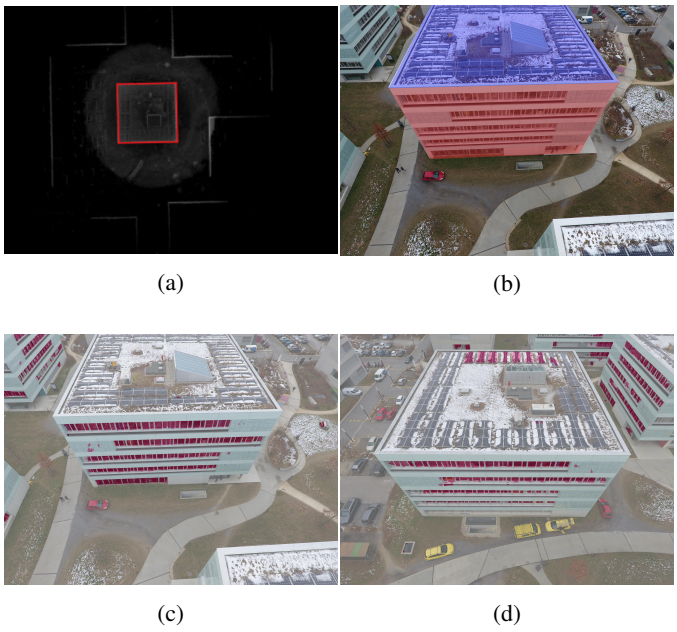
Results

Using the 85 drone images and Pix4D photogrammetry software a 3D point cloud was generated, and was used to extract the building's 3D geometry (see section 3.1). Figure 3(a) shows a top down view of the the point cloud with the detected building footprint. Figure 3(b) shows the projection of the detected building's 3D geometry on top of a drone image at the East side of the building. For each side of the building the drone image that has the most frontal view of the facade was selected and used as an input to the trained DeepLabv3+ to generate windows masks. The segmentation accuracy of each of these masks along the four facades are shown with the F1 score in Table 1. One can see that only one of the 4 facade has an accurate result in term of F1 score, which is mainly due to the angle of view of the camera (i.e., drone and camera position relative to the facade). Figure 3(c) show the east facade drone image overlaid with the detected windows mask, and Figure 3(d) show the north facade with the corresponding mask. It is clear that the angle of view of the camera from the east facade provides a better view of the windows, which explains the more accurate results for this facade. Meanwhile, for the northern facade the drone was too close to the building to capture a good view of the windows, especially for the first two floors of the building. Similar issues are present for the western and southern facades. WWR were estimated following the previously described method. Table 1 shows the estimated WWR and the ground truth WWR (noted as WWR in the table). As expected, the most accurate results are obtained for the east facade.

[2]https://support.pix4d.com/hc/en-us/articles/360000235126-Example-projects#label6

Table 1: Accuracy metrics

Facade	F1 Score	estimated WWR	WWR
North	0.73	0.16	0.23
East	0.88	0.21	0.23
South	0.67	0.12	0.24
West	0.66	0.13	0.24



(a) (b) (c) (d)

Fig. 3: (a) Top-down view of the point cloud with the detected building footprint; (b) 3D building model projected onto a 2D drone image, with roof polygon overlaid in blue and facades in red; (c) an image of the eastern facade with the detected windows overlaid in red; (d) an image of the northern facade with the detected windows overlaid in red

5. CONCLUSIONS

We presented a novel machine learning approach to estimate the WWR using drone imagery.We evaluated our semantic segmentation model choices based on an F1 score and show experimentally that our models can achieve accurate performance, by comparing our estimated WWR with actual WWR number. A limitation of our approach is that the accuracy of the WWR depends on the drone and camera position relative to the facade. In the future, this approach can be extended for parsing additional building categories such as doors, balconies and rooftop PV.

6. REFERENCES

[1] Markus Mathias, Andelo Martinovic, Julien Weissenberg, Simon Haegler, and Luc Van Gool, "Automatic architectural style recognition," in *Proceedings*, 2011, vol. 38, pp. 171–176.

[2] Andrea Cohen, Alexander G Schwing, and Marc Pollefeys, "Efficient structured parsing of facades using dynamic programming," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3206–3213.

[3] Olivier Teboul, Iasonas Kokkinos, Loic Simon, Panagiotis Koutsourakis, and Nikos Paragios, "Shape grammar parsing via reinforcement learning," in *CVPR 2011*. IEEE, 2011, pp. 2273–2280.

[4] Leon A Gatys, Alexander S Ecker, and Matthias Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015.

[5] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pp. 240–248. Springer, 2017.

[6] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[8] Matthias Schmitz and Helmut Mayer, "A convolutional network for semantic facade segmentation and interpretation.," *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 41, 2016.

[9] SM Iman Zolanvari and Debra F Laefer, "Slicing method for curved façade and window extraction from point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 119, pp. 334–346, 2016.

[10] Sung Chun Lee, Soon Ki Jung, and Ram Nevatia, "Automatic integration of facade textures into 3d building models with a projective geometry based line clustering," in *Computer Graphics Forum*. Wiley Online Library, 2002, vol. 21, pp. 511–519.

[11] Sung Chun Lee and Ramakant Nevatia, "Extraction and integration of window in a 3d building model from ground view images," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* IEEE, 2004, vol. 2, pp. II–II.

[12] Hoang-Hon Trinh, Dae-Nyeon Kim, Suk-Ju Kang, and Kang-Hyun Jo, "Window extraction using geometrical characteristics of building surface," in *International Conference on Intelligent Computing*. Springer, 2009, pp. 585–594.

[13] Marc WuDunn, Avideh Zakhor, Samir Touzani, and Jessica Granderson, "Aerial 3d building reconstruction from rgb drone imagery," in *Geospatial Informatics X*. International Society for Optics and Photonics, 2020, vol. 11398, p. 1139803.

[14] Radim Tyleček and Radim Šára, "Spatial pattern templates for recognition of objects with regular structure," in *Proc. GCPR*, Saarbrucken, Germany, 2013.

[15] F. Korč and W. Förstner, "eTRIMS Image Database for interpreting images of man-made scenes," Tech. Rep. TR-IGG-P-2009-01, April 2009.

[16] F Nex, F Remondino, M Gerke, H-J Przybilla, M Bäumker, and A Zurhorst, "Isprs benchmark for multi-platform photogrammetry.," *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 2, 2015.

[17] 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)*, 2018, pp. 801–818.

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.