Towards automatic construction of photorealistic BIM window elements from single-picture input

Johan Luttun, johan@aecgeeks.com
AECgeeks, Eindhoven, Netherlands

Thomas Krijnen, t.f.krijnen@tudelft.nl
Delft University of Technology, Delft, Netherlands

Abstract

Recent progress in computer vision and the pervasive use of Building Information Models (BIM) in the construction industry provide countless opportunities for decision making in architectural design. It allows capturing real-world geometrical data for assessing design options, such as windows products for replacement on the façade of an existing building. However, capturing information from the real world is a tedious task that involves data acquisition followed by costly processing steps. In this study, we focus on using a single 2D picture containing simple building elements in their context to generate a semantic 3D model of these elements. To that aim, we trained a detection neural network whose input constituted of labeled photographs containing windows. After training, the model is able to locate and segment elements of interest on unseen new data. With the use of Computer Vision methods, we derive accurate geometric outlines suitable to reconstruct photorealistic three-dimensional BIM models.

Keywords: BIM, VR, VR environment, Computer Vision, Object detection, Photorealistic BIM

1 Introduction

Refurbishing the existing buildings is a necessary challenge for sustainability (Abergel et al., 2018), and in the context of building renovation design, replacement of the windows accounts for an important contribution to the future thermal performance, environmental footprint, and aesthetics of the building they are inserted into (Galimshina et al., 2020). Working as a physical membrane between the interior and exterior spaces, providing protection against the weather, windows also have the function of letting the ray lights pass through their glazing components, offering lighting, luminance, solar radiation, and a view to the outside (Jelle, 2013) and thus are essential to any construction by playing a significant role in the well-being of users (Galimshina et al., 2020).

Making a choice for windows replacement necessitates to consider physical and nonphysical contradictory properties. Regarding its intended physical behavior, a window should provide as much daylight as possible and reduce thermal loss to the outside but also transmit or reflect solar radiation to lower the energy demand or to avoid overheating. Therefore, formulating a design choice when dealing with windows replacement involves to carefully choose a product with respect to these physical properties, which is dependent on the window's frame material and its glazing (Jelle, 2013). Concurrently, windows are also assessed in terms of aesthetics and visual comfort as they are part of building façades, which makes its selection even more difficult.

Creating 3D models that can serve as a starting point to make decision design of new objects (Bernardini & Rushmeier, 2002) like windows requires time-consuming and error prone manual or semi-automatic data acquisition, data processing, 3D reconstruction and parametric modeling tasks (Macher et al., 2017). With the advent of Building Information Modeling, it is possible to model, annotate, and share a model of an existing building façade containing the windows to the various stakeholders of a refurbishment project. The input data to produce such BIM can consist in individual or combined representations of the real world like 2D plans (Vosselman & Dijkman, 2001), photographs (Braun et al., 2015) or point clouds (Son et al., 2014), that are further processed through a pipeline to obtain a digital representation of the building.

Many studies have been conducted to reduce the labor and computational costs of the 3D reconstruction process by automating some of its steps. In those studies, the common pipeline consists in (1) acquiring point cloud either from LIDAR techniques or from photogrammetry, (2) segmenting the point cloud according to rules derived from expert knowledge, (3) reconstruct the geometry of the elements, and (4) assign semantic meaning to the reconstructed elements. Even if those studies have contributed positively towards solving the 3D reconstruction problem, they remain expensive in terms of processing time (Benarab et al., 2018) or devices (Bernardini & Rushmeier, 2002), and they need to hardcode geometric rules to segment the point clouds. Regarding those difficulties, the 3D reconstruction process must be adapted to the objects to represent, the budget available, and the objective of the representation.

Current progress in computer vision and machine learning brings promising opportunities to tackle those challenges and offer a more uniform and lighter framework to represent the world digitally. The access to more efficient hardware devices and the proliferation of data on the worldwide web enables to train supervised machine learning models that can learn the mapping from input data to output target. Thanks to this technique, some of the 3D reconstruction process steps can be eased with the use of detection models like in (Behzadan, 2020), where they used the machine learning approach to detect objects present in the building site as an alternative to LIDAR or photogrammetry.

However, there still exists barriers for the use of machine learning techniques, such as the data acquisition, the hardware availability, and the annotation step. (Behzadan, 2020) research required an effort on the data acquisition and the labeling part necessary to train a supervised neural network. To ease these challenges, the emergence of transfer learning enables to use open-sourced pretrained model on huge datasets and performant hardware with new data (Weiss et al., 2016). The advantage is that in the case of supervised learning tasks, which require annotated dataset, the labeling part is reduced, and small datasets are enough to obtain satisfying results.

One of the most important studies working on the 3D reconstruction problem uses the Pointnet neural network in a generative way (Fan et al., 2017). In this work, the authors, use 3D point clouds to represent a 3D shape. They use Shapenet (Chang et al., 2015) models repository as ground-truth and create virtual screenshots of these models. They train their network to generate 3D point clouds and minimize loss over the ground-truth. While training on picture of 3D model, they manage to achieve some interesting results on real world pictures.

However, even if the 3D models' stock is growing, there is still a lack of ground-truth references for objects in the context of specific industry-related task detection. Existing 3D models datasets have been created and used to supply studies mostly focusing on reconstructing daily life objects for indoor scene reconstruction[1] or human and body reconstruction[2].

More recently self-supervised learning method from (Li et al., 2020) aims at reconstructing different pictures without ground-truth references. The underlying process of their method is to describe common semantic attributions across images of the same nature. This research provides promising opportunities for segmenting pictures with a lowered labeling effort.

Our research aims to provide answers on the two following questions:

- How can data acquisition effort be reduced for visually assessing building retrofits?

[1] 3D-FRONT. (n.d.). 3D-FRONT. Retrieved May 5, 2021, from https://tianchi.aliyun.com/specials/promotion/alibaba-3d-scene-dataset [last accessed May 2021]

[2] MPII Human Shape. (n.d.). MPII Human Shape. Retrieved May 5, 2021, from https://humanshape.mpi-inf.mpg.de/ [last accessed May 2021]

- What are the potential and challenges on using merely a simple camera for scene reconstruction for this purpose?

We take advantage of the simplicity of the shape we wish to reconstruct by hardcoding the construction of the 3D window model. As our study is limited to rectangular windows embedded in walls via a rectangular void, we focus on detecting windows voids. We are also training on all shapes to make our network reusable. We use a neural network to detect and segment windows instances and then reconstruct them in an enriched 3D format. Our method provides a benchmark for the reconstruction of other objects than the windows. The advantage of our methods lies in the fact that it does not need a lot of data to train, nor powerful hardware.

2 Methods

2.1 Windows detection

2.1.1 Detection model used

We used the opensource detectron2 library to train a detection and instance segmentation model that can detect windows voids instances. The detectron2 library provides some utility function to prepare the datasets, train models, and evaluate their performance. Among those models, we used the Mask R-CNN architecture developed in (He et al., 2018).

This model is an improvement of the Fast CNN architecture, and is based on two parts: a Region Proposal Network, and a classifier. Mask R-CNN is an extension of Faster R-CNN, detection neural network, in which a module has been added to segment the region of interest. The segmentation part consists in a Fully Convolutional Network (FCN) that maps every pixel of the region of interest to a category.

Detectron2 enables to use pretrained models that were trained on thousands of images. Thanks to this transfer of learning, we can train our model with only hundreds of examples. This lower the barrier to using machine learning algorithms by reducing the data acquisition effort, the labeling effort, and without requiring the use of powerful hardware. While the model was trained on different objects than windows voids, the model learned how to characterize patterns on the objects on pictures that are shared across pictures containing similar or different instances.

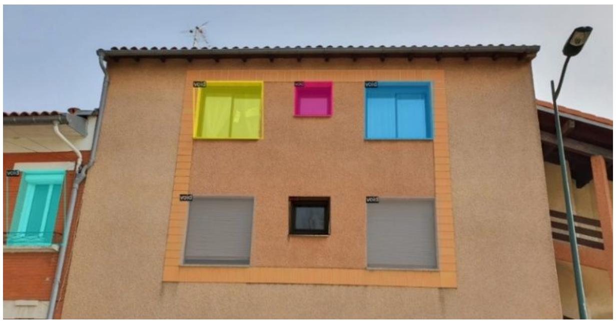

Figure 1. Annotated picture with bounding box and mask for every window void

2.1.2 Experimental set up

We labeled 476 windows voids from 126 real-world photographs or pictures from the worldwide web. As shown in Figure 1, the labeling part consists in creating for each window void instance a polygonal mask that is defined by its vertices and to assign a class to this mask.

We made the choice to label the voids instead of the windows objects because we need the void corners to create the extrusion in the wall.

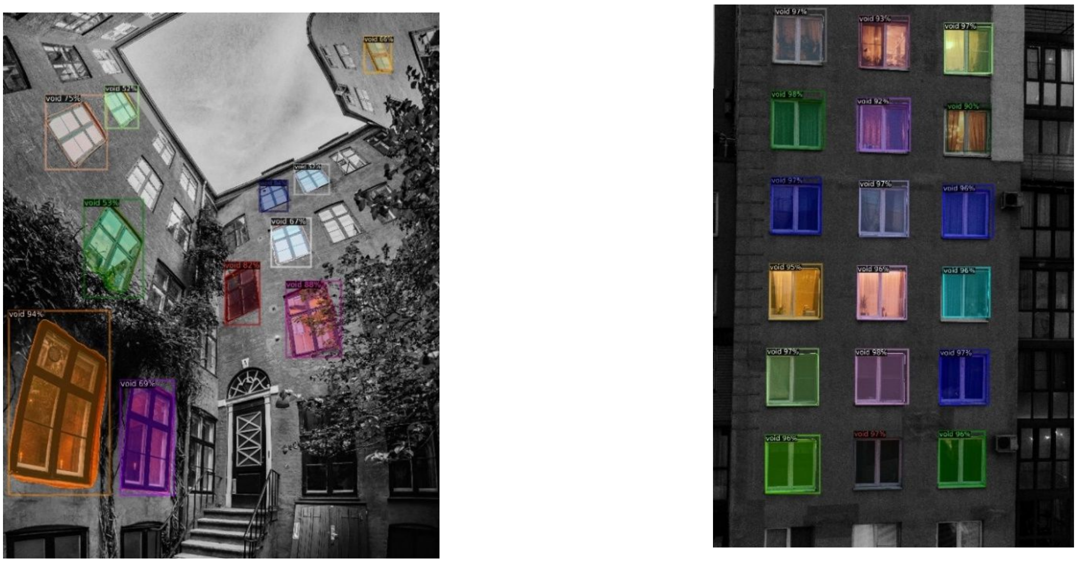

Figure 2. Illustrations of the windows voids detected on unseen new data. The model is able to detect windows different shape and from different angles.

2.2 Windows reconstruction

2.2.1 Region mask clean-up

The 3D reconstruction used in our research depends on a correspondence between 2D points on the image and 3D points in BIM. Detectron2 gives us a region mask with arbitrary shape and even for a detected quadrilateral will be somewhat irregular at the edges. Therefore, at every point of the polygon mask we create an infinite line along the principal component direction fitted through 100 neighboring points. We pick the rays that explain most points, and keep their intersection points, which in case of conventional windows yields a two-dimensional quadrilateral.

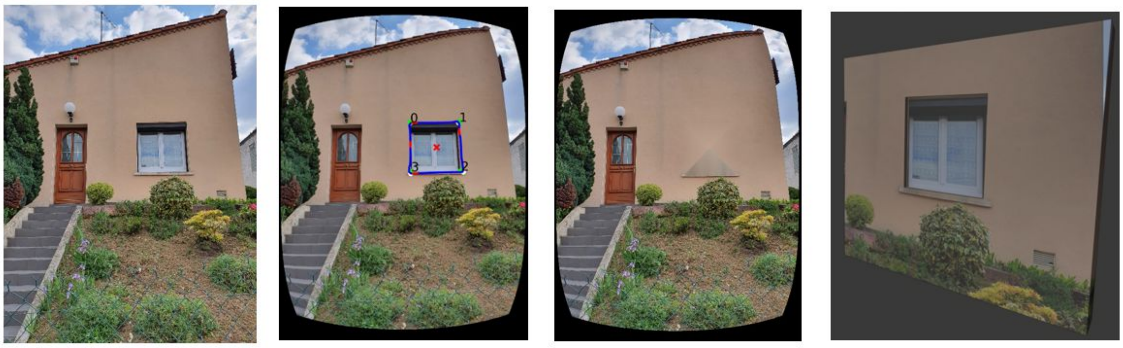

Figure 3. Visualization of the reconstruction pipeline final and intermediate results.

This is purely a geometric method that disregards information from the image pixels. For that reason, we create a neighborhood around the four window corners and use the Harris (Harris & Stephens, 1988) points in the OpenCV computer graphics library to “snap” the geometrically constructed points back to strong corners in the image. This is illustrated in Figure 4.

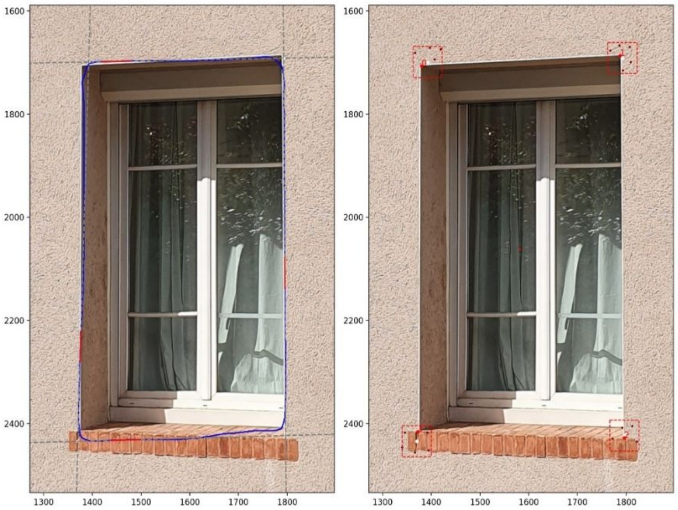

Figure 4. Illustration of the detection mask clean-up process with on the left in blue the original mask, in red the four tangents that explain most mask vertices, their extension in gray. On the right the intersection of these extensions is displayed in white with a search area around them in which several Harris corners are identified, in bright red the closest to the search area center is indicated, which is selected and indeed corresponds to the corner of the actual window void.

2.2.2 Aspect ratio

The pose estimation employed in our research (see below) depends on a correspondence between 2D and 3D points, but this correspondence is invariant of scaling (size and distance are inversely proportional to one another, in future research we will investigate incorporating learning approaches to estimate distance fields such as (Haseeb et al., 2018)). As such, for a rectangular window void, all four windows points can be characterized by means of a single value: the aspect ratio, the ratio between width and height of the window frame. In this research we rely on (Zhang & He, 2007) to derive the aspect ratio from the perspective projection of the four points in two-dimensional image coordinates.

2.2.3 3D reconstruction

The two-dimensional quadrilateral is transformed into three dimensions with pose and translation using the pose estimation functionality in OpenCV using the iterative Levenberg-Marquardt least squares error optimization. The 3D window quadrilateral is then transformed into IFC procedural solid geometry definitions with a wall fragment around it, both taking texture coordinates from the 2D image. An instance graph of the generated IFC output is shown in Figure 5 with additional explanations in Table 1 on how the values are derived. The accuracy of the 3D reconstruction depends heavily on the accuracy of the 2D-3D correspondence and hence on the estimation of aspect ratio. The estimation of the aspect ratio depends on a properly calibrated camera (a fairly straight forward process that involves taking pictures of chessboard prints and automated analysis thereof) to eliminate any potential lens distortion. It also depends on an accurate positioning of the four corner points to correctly represent the perspective projection, which is why the Harris feature detection part is crucial in the current setup.

2.2.4 Texture map

The enhanced window mask is erased from the image by means of inpainting. For the version presented in this paper we relied on (Telea 2014) as implemented in OpenCV, but likely better results can be obtained by means of a learning approach such as (Iizuka, 2017).
Since texture support is not found in most of the current IFC model viewers, we relied on a procedurally created IFC file, which is immediately converted to the Wavefront OBJ file format by means of the IfcConvert[3] executable. This OBJ file is augmented with texture coordinates which are simply transformation of the vertex positions taking into account camera matrix and pose and translation vectors obtained from the OpenCV pose estimation.

2.2.5 Windows substitution

This is a similar idea as presented in (El-Diraby et al. 2017) where the user can select an element from a catalogue and a modification is made to the IFC instance graph. Surface styles and colors are taken from the original IFC file. The wall is rendered in a way that it does not take into account lighting, but for the window an artificial light source is place into the scene. Although applied to portraits, something akin to (Pandey et al. 2021) might help in matching the lighting from the original image with the newly substituted window. Figure 6 shows how by overlaying the newly substituted window, embedded in the wall fragmented, a visual indication of the replacement can be indicated to users using a "photorealistic BIM" approach. No manual interaction was needed.

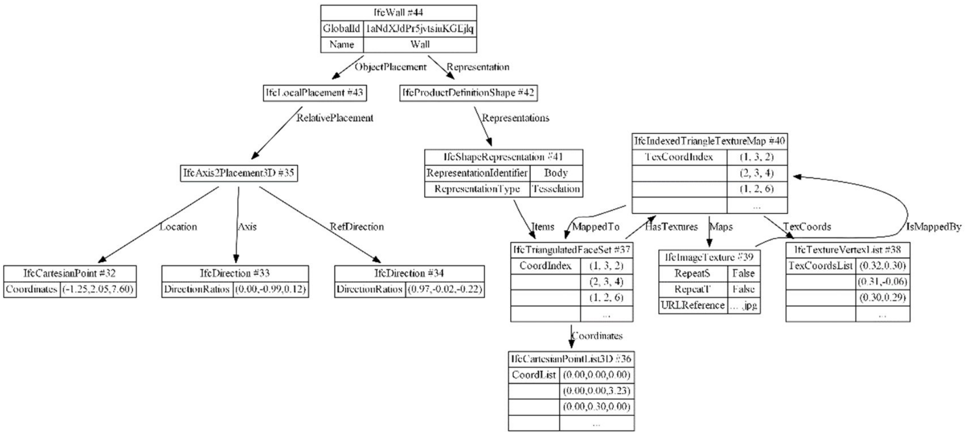

Figure 5. Instance graph of the generated IFC data.

Table 1. Explanations for the generated data laid out in Figure 5.

IFC instance ID	Description
#32	The placement location of the wall is computed as a 1m X,Z offset from the lower corner of the Window. The Window position itself includes the translation vector obtained from the pose estimation.
#33, #34	The placement directions of the wall are obtained from the pose estimation. This means that the wall is rotated in space. The reason is that the IFC standard does not offer ways to include a camera and we want to be able to reproject the wall with substituted window back onto the original image. In this setup the camera is at (0, 0, 0)
#36	The local coordinates of the wall, which would normally be an extrusion, but has to be tessellated for IFC4 to support the association of texture coordinates. The length and height are derived from the window size plus 1m margin. The 30cm thickness is currently hardcoded.

[3] IfcConvert: An application for converting ifc geometry into several file formats. (n.d.). IfcOpenShell. Retrieved May 5, 2021, from http://ifcopenshell.org/ifcconvert [last accessed May 2021]

#38	The UV coordinates of the texture which are obtained by camera matrix × wall placement matrix × local coordinates.
#37, #40	Since the UV texture coordinates are direct transformations of the XYZ wall positions, the indices are identical.

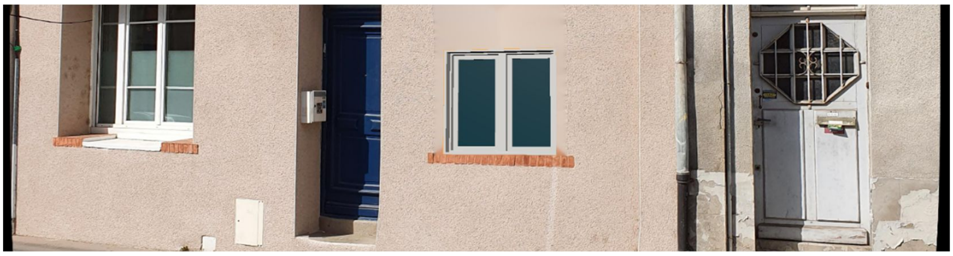

Figure 6. Illustration of the window replacement overlaid on top of the original image. Inpainting artefacts are visible near the windowsill. Lighting of the window frame has been manually adjusted.

3 Results

3.1 Detection accuracy

In this section we highlight the accuracy of the machine learning part to detect window frames in single picture input and, in the section, thereafter, discuss the accuracy of the pose estimation and 3D reconstruction by means of a synthetic dataset.

3.1.1 Comparison on similar experiment

The dataset was split as follow: 88 images in our training set, containing in total 380 voids and 38 images in the testing set, containing in total 96 voids. The bounding box classification layer outputs a vector with the probability of each category. In our case, we only have the void category. We can control the classification threshold which we set to 0.5, which means, that any prediction with a classification confidence metric less than 0.5 will not be retained. We show an example of a prediction made in Figure 7a for a lowered threshold of 0.1.

The performance of Mask R-CNN architecture can assessed with the Average Precision (AP) metric (He et al., 2018). This can be measured for detection and segmentation tasks. The AP metric reflects the accuracy of the classification and of the spatial precision of the detections made by the model. We used the COCO evaluator to calculate the AP score on our testing set[4].

The AP score takes into account the Intersection over Union ratio which represents the portion of the prediction that represents the ground-truth. The AP score uses the Precision score. The precision measures the accuracy of the prediction, that is the number of correct voids predictions among the total number of predictions. In addition, the Recall score measures the model ability to find the correct predictions over all the possible voids. We refer the reader to COCO (Common Objects in Contexts) for more details about the computation of these metrics of the AP scores by the COCO evaluator.

As shown in Table 2, we obtained competitive AP scores of 61.9 for the detection task and 72.48 for the segmentation task. Comparatively, in (Behzadan, 2020), where they do object detection for building sites, they used 2507 images containing 11572 instances of 3 categories, labeled with bounding boxes only. They obtain at best on their 3 categories an average AP of 77.3 So they are a little bit less than 30% better than us on AP for detection, with almost twenty times more annotated images. However, we acknowledge that a fair comparison should involve models testing on different dataset size with increased variety of categories.

[4] COCO - Common Objects in Context. (n.d.). COCO - Common Objects in Context. Retrieved May 5, 2021, from https://cocodataset.org/#detection-eval [last accessed May 2021]

Table 2. COCO evaluation results on the testing set

Task	AP	AP50	AP75	APs	APm	APl
Detection	61.90	92.763	81.066	15	58.914	63.812
Segmentation	72.476	92.763	88.715	30	64.560	75.195

3.1.2 Prediction examples

To get a better understanding of how the detection model performs and the flaws it can have, Figure 7a shows in more details some examples of the predictions made. First, we can see that 7 bounding boxes (classification true positives), were accurately detected on the first floor of the 3 buildings in the picture. However, the map on the bus stop was detected with 66% confidence, which gives a classification false positive. The same happens for the circular ornament above the left most window, which is detected as a window's void with a 54% confidence, and for the rectangular element on the right side of the right most door. In addition, we can see that the model was not able to predict the 2 right most windows, as it only has 22% and 34% confidence, which gives us classification false negatives. On the other hand, on Figure 7b we can see an example of the superposition between the ground-truth bounding box and the predicted one with an IoU of 0.86.

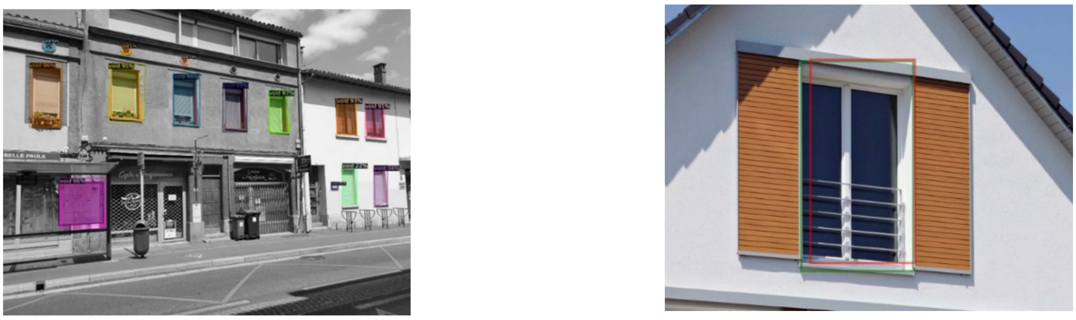

Figure 7. (a) Prediction on a new picture with a lowered output threshold of 0.1 and (b) Illustration of the overlay between the ground-truth bounding box (green) and the predicted one (red).

3.2 Reconstruction accuracy

The work presented in this paper consists of two parts. Machine learning for the detection and segmentation of window void regions and after that a set of computer vision and reconstruction approaches to derive accurate three-dimensional window frames from the masks. The accuracy of the machine learning can easily be tested as we have manually labeled images and a test set. To understand the accuracy of the computer vision part we have resorted on a rendered BIM model [5]. We automatically created screenshots containing windows at various orientations. For each screenshot, we took the real window void corner points and projected them to two dimensions using the camera matrix. We ran the detection and then reconstruction logic on the same synthetic building model screenshots and took note of three sets of points that are used throughout the algorithm. These are displayed in Figure 8. The mask tangent intersection point is derived from only the segmentation mask. While the maximum error distance is smallest for this set of points, the histogram for the Harris corner points clearly shows that the frequency of small errors (<5 pixels) is higher. The same histogram also hints as a

[5] Department of Computer Science, University of Auckland. (n.d.). House Sr. Dubal Herrera. Open IFC Model Repository. Retrieved June 26, 2021, from http://openifcmodel.cs.auckland.ac.nz/Model/Details/314 [last accessed June 2021]

multimodal distribution. This can be explained from the fact that the Harris corner algorithm has the tendency to snap to the sills at the bottom of the window, which projects outwards from the opening so that depending on orientation, can be several pixels away from the opening corner. The last histogram shows that the error is enlarged by the pose estimation, even in case of a perfect perspective camera without lens distortion.

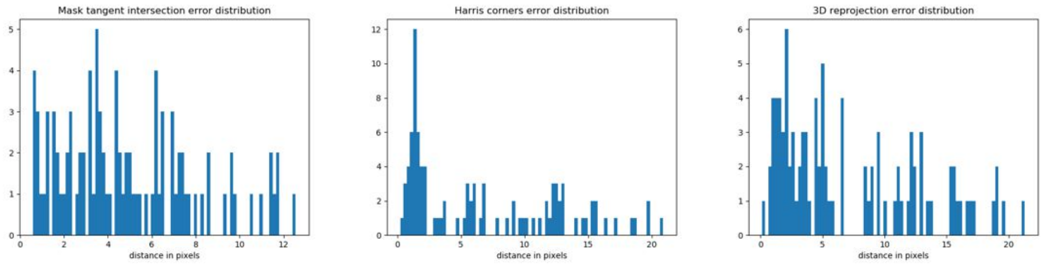

Figure 8. Error distributions of reconstruction and computer vision.

Conclusion

In this paper we demonstrated a successful approach to generate photorealistic BIM models of rectangular windows from single-picture input using a simple calibrated smart phone camera. The approach depends on machine learning for object detection and a following step of computer vision to fine tune the segmentation mask into precise quadrilateral for pose estimation and 3D reconstruction.

This work presents an isolated application of learning and computer vision without interaction between the two. As such, one of the limitations of our approach, is that we based the 3D reconstruction from detection for rectangular windows only, whereas we have the opportunity to detect with a competitive performance the voids for any shape. To broaden our reconstruction to any polygonal shaped void, it would be interesting to follow an “analysis by synthesis” scheme, which would train a model to generate 3D proposition until it matches the input data and supplemental hardcoded rules. Such an approach would also have the advantage to learn an 2D to 3D mapping invariant to the user's camera intrinsic and extrinsic parameters, making the overall process even easier.

In addition, we do not infer the depth of the void in our method, as we are setting a default width of the embedding wall to 30 cm. We could overcome this lazy assumption by supplementing our detection model with keypoint detection in addition to bounding box and mask segmentation, to get the skeleton of the 3D structure of the void and the window. The exact scale of the scene is also something that is currently still set with a single manual parameter.

Another flaw of our method is the classification errors in the case of complex scene or complex façades as it was shown in Figure 7a. We are confident that an increase in the number of configurations in our dataset could solve in great part this issue. We initiated this reflection by classifying input picture according to several parameters such as the number of windows, the type of picture (scene, façade only, or portrait). In addition, we could also experiment various splitting modes regarding the labels by distinguishing rectangular from polygonal voids.

Our contribution wishes to propose a method to lower the barrier to the data acquisition, reconstruction, and visualization of existing building elements. This work also attempts to bring latest advance from the vibrant machine learning and computer vision communities to an industry that more than ever needs a fast, reliable, and more affordable way to model the built environment and edit with design choices that will have a long-lasting positive impacts.

We show that we can successfully detect windows from single pictures after limited training, so this work can be used to detect and reconstruct other elements such as the doors. However, another difficulty would thus emerge as the doors are often occluded by vehicles, urban furniture, or persons in a street scene setting. Especially the computer vision feature detection would need to be made more robust as it would otherwise snap window corners to the high contrast areas introduced by the occlusions. Hough transforms seem like a reasonable first effort at this by prioritizing lines in the model as opposed to just points with high contrast.

References

Abergel, T., Dean, B., Dulac, J., Hamilton, I., & Wheeler, T. (2018). *Global Status Report—Towards a zero-emission, efficient and resilient buildings and construction sector*. International Energy Agency and the United Nations Environment Programme. https://www.worldgbc.org/sites/default/files/2018%20GlobalABC%20Global%20Status%20Report.pdf

Behzadan, A. H. (2020). Deep Convolutional Networks for Construction Object Detection Under Different Visual Conditions. *Frontiers in Built Environment*, *6*, 22.

Benarab, D., Derigent, W., Brie, D., Bombardier, V., & Thomas, A. (2018). All-Automatic 3D BIM Modeling of Existing Buildings. In P. Chiabert, A. Bouras, F. Noël, & J. Ríos (Eds.), *Product Lifecycle Management to Support Industry 4.0* (Vol. 540, pp. 56–68). Springer International Publishing. https://doi.org/10.1007/978-3-030-01614-2_6

Bernardini, F., & Rushmeier, H. (2002). The 3D Model Acquisition Pipeline. *Computer Graphics Forum*, *21*(2), 149–172. https://doi.org/10.1111/1467-8659.00574

Braun, A., Tuttas, S., Borrmann, A., & Stilla, U. (2015). A concept for automated construction progress monitoring using BIM-based geometric constraints and photogrammetric point clouds. *Journal of Information Technology in Construction (ITcon)*, *20*(5), 68–79.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., & Yu, F. (2015). ShapeNet: An Information-Rich 3D Model Repository. *ArXiv:1512.03012 [Cs]*. http://arxiv.org/abs/1512.03012

Fan, H., Su, H., & Guibas, L. (2017). A Point Set Generation Network for 3D Object Reconstruction from a Single Image. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2463–2471. https://doi.org/10.1109/CVPR.2017.264

Galimshina, A., Moustapha, M., Hollberg, A., Padey, P., Lasvaux, S., Sudret, B., & Habert, G. (2020). Statistical method to identify robust building renovation choices for environmental and economic performance. *Building and Environment*, *183*, 107143. https://doi.org/10.1016/j.buildenv.2020.107143

Harris, C., & Stephens, M. (1988). A Combined Corner and Edge Detector. *Procedings of the Alvey Vision Conference 1988*, 23.1-23.6. https://doi.org/10.5244/C.2.23

Haseeb, M. A., Guan, J., Ristić-Durrant, D., & Gräser, A. (2018). *DisNet: A novel method for distance estimation from monocular camera*. 6.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2018). Mask R-CNN. *ArXiv:1703.06870 [Cs]*. http://arxiv.org/abs/1703.06870

Jelle, B. P. (2013). Solar radiation glazing factors for window panes, glass structures and electrochromic windows in buildings—Measurement and calculation. *Solar Energy Materials and Solar Cells*, *116*, 291–323. https://doi.org/10.1016/j.solmat.2013.04.032

Li, X., Liu, S., Kim, K., De Mello, S., Jampani, V., Yang, M.-H., & Kautz, J. (2020). Self-supervised Single-View 3D Reconstruction via Semantic Consistency. In A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Eds.), *Computer Vision – ECCV 2020* (Vol. 12359, pp. 677–693). Springer International Publishing. https://doi.org/10.1007/978-3-030-58568-6_40

Macher, H., Landes, T., & Grussenmeyer, P. (2017). *From Point Clouds to Building Information Models: 3D Semi-Automatic Reconstruction of Indoors of Existing Buildings*. 31.

Son, H., Kim, C., & Kim, C. (2014). Automatic 3D Reconstruction of As-Built Pipeline Based on Curvature Computations from Laser-Scanned Data. *Construction Research Congress*, 10.

Vosselman, G., & Dijkman, S. (2001). 3D BUILDING MODEL RECONSTRUCTION FROM POINT CLOUDS AND GROUND PLANS. *International Archives of Photogrammetry and Remote Sensing*, 37–43.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, *3*(9). https://doi.org/10.1186/s40537-016-0043-6

Zhang, Z., & He, L.-W. (2007). Whiteboard scanning and image enhancement. *Digital Signal Processing*, *17*(2), 414–432. https://doi.org/10.1016/j.dsp.2006.05.006