System for Detection and Tracking of Windows in Urban Environment *

Krzysztof Krygiel[1][0000−0003−0017−0299], Karol Majek[1][0000−0002−1351−8496], and Janusz Bedkowski[2][0000−0003−2630−1947]

[1] Mobile4.pro sp. z o.o., 02-621 Warsaw, Poland;
karol.majek@mobile4.pro
[2] Institute of Fundamental Technological Research, Polish Academy of Sciences, 02-106 Warsaw, Poland

Abstract. In this paper, we propose a system for tracking window edges, which can be used to improve computer vision tasks such as visual odometry in urban scenes. We use deep neural network based detector to detect windows as aligned bounding box predictions. In the second step, we track windows using keypoint-based tracking. We analyze the graph of detected matches across multiple frames to remove outliers. In the final step, we use learned line detector to refine the axis aligned bounding box approximation of the window. Each component of the system can be improved to achieve better results of the system.

Keywords: Window detection · tracking · image matching.

1 Introduction and Related Work

In urban environments such as large cities, repeating patterns such as windows occur frequently. In multi-store apartment buildings windows are tessellated which could lead to assignment errors in computer vision methods such as visual odometry or structure from motion. In this paper, we focus on apartment windows in particular and propose a system which can detect, track and refine detected windows to provide reliable information across multiple frames.

Window detection can be performed from aerial images using instance segmentation technique [5] as well as from ground imagery [9]. It is one of the tasks in building facades parsing task [7] which is currently solved with semantic segmentation and instance segmentation, deep learning based methods [9]. Windows are annotated in Open Images dataset [3] and for the purpose of experiments in this work we use a pretrained model on this dataset by [10]. Several methods can be used for tracking detected windows starting from tracking by detection using Intersection over Union (IoU) with boxes from subsequent frames requires

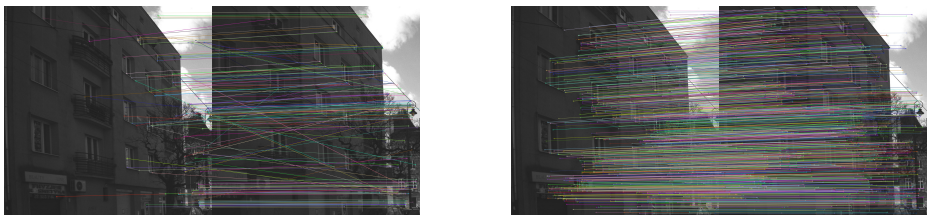
* The research leading to these results has received funding from POIR.01.01.01-00-0494/20 "Development and verification of the automatic location and 3D visualization of the selected objects in urban environment technology together with people flow modeling".

a guarantee of small movements of objects in the image [1]. Image matching state of the art was significantly improved by SuperPoint network [2] compared to hand-crafted detectors and descriptors. This feature matching deep front-end was later improved by introducing deep middle-end matcher SuperGlue [12]. It introduced Graph Neural Networks to solve the assignment optimization problem. Recently, a detector-free method was introduced to solve image matching - LoFTR [13] which uses Transformer architecture with global receptive field to provide dense matches at coarse level and refine only good ones.

In our work we do not aim to parse facades accurately, instead we want to provide good matches between images from cities where tessellated structures such as windows occur. Therefore, we propose a modular system for detection, and tracking of windows.

2 Proposed System

We propose a system which consists of 4 steps: axis-aligned bounding box detection, tracking using image matching, graph-based outlier removal, and finally refinement based on-line detection. In first step we perform detection using axis aligned window detection method based on YOLOv3 [11] trained on Open Images dataset [3]. In the second step we aim to match windows in consequtive frames. Our approach is to use keypoints-based image matching methods in order to solve association problem. Methods such as *ORB* or *SIFT* turned out to be not precise enough to track windows as shown in Figure 1a. We confirmed that the *SuperPoint* method for generating characteristic points and *SuperGlue* [12] for connecting them is sufficient in our case. Figure 1b presents *Superglue* result for the same pair of pictures.



(a) SIFT [8] (b) Superpoint [2] + Superglue [12]

Fig. 1: Keypoint matching results for object detection task; hand-crafted detector compared to learned method

The goal of window tracking is to assign an unique identifier to each physical window. Therefore, errors can be of two types: same identifier may by assigned to different windows, single window observed in many photos may receive more than one identifier. In order to detect which matches are invalid, it is worth of transforming that problem using graph theory. Each vertex is a single bounding

box detected by the neural network. Edges connect vertices only when bounding boxes were matched, considering an appropriate number of matching key points as a condition. If all matches were correct, then the graph would contain many connected components and each component would represent a single physical window.

Such component should contain a lot of edges from single vertex because single bounding box should match with corresponding boxes from neighboring frames. Wrong matches are converted to edges, which create a single component from two components which should remain separate. If there is just one such edge, it is called a *bridge* - edge of a graph whose deletion increases the graph's number of connected components. It is possible to find bridges in the graph to eliminate invalid matches. Due to the fact that errors might be a bit more frequent it is good to look for components such that even if 2 edges are removed it still remains a component. Each *3-edge-connected component* should contain all vertices representing bounding boxes from different images with same physical window inside. After 3 steps of the system, the result is presented in Figure 2. In the last step, we apply line detection method such as Deep Hough-Transform Line Priors [4] or HAWP [14] to refine the bounding boxes.

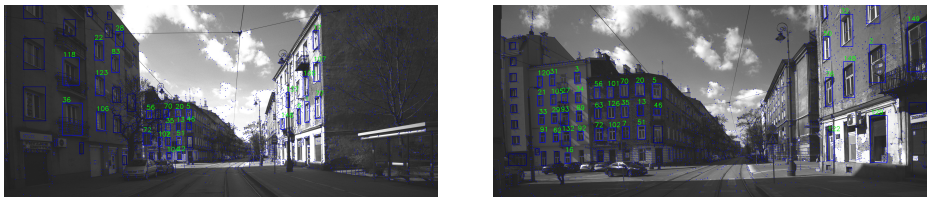


Fig. 2: Preserved window identifiers using keypoint-based tracking in two frames captured from distant positions with assigned identifiers. Axis aligned window detection method based on YOLOv3 [11] trained on Open Images dataset [3] from [10]

3 Conclusion and Future Work

We presented a system for detection and tracking window edges. Proposed system can track similar objects reducing the number of mismatched windows thanks to used graph representation. We identified two problems which are solved thanks to image matching: tracking obscured windows invisible, and not detected in several frames, and when there is an object which partially covers the window in one frame and another window in the second frame, there might exist enough matches between points on that obscuring object and those bounding boxes may be connected. Thanks to the modularity of the proposed system, we identify multiple ways of further improving the results of detection, assignment, and tracking of windows. Detection module can be replaced with

recent state-of-the-art method [6]. Window edges detection may be improved by preparing dataset containing images captured from the city and using them to train network detecting wireframes. The system output can be improved when use LoFTR image matching [13]. Tracking module can be improved with recent methods such as [15]. The interesting next step would be to use Graph Neural Networks such as in [12] to eliminate bad matches. Evaluation of the proposed system on a visual odometry task on publicly available datasets is planned a future work, as well as involving multi object tracking metrics.

References

1. Bochinski, E., Eiselein, V., Sikora, T.: High-speed tracking-by-detection without using image information. In: 2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS). pp. 1–6. IEEE (2017)
2. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 224–236 (2018)
3. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T., Ferrari, V.: The open images dataset v4. International Journal of Computer Vision **128**(7), 1956–1981 (3 2020). https://doi.org/10.1007/s11263-020-01316-z, http://dx.doi.org/10.1007/s11263-020-01316-z
4. Lin, Y., Pintea, S.L., van Gemert, J.C.: Deep hough-transform line priors (2020)
5. Lippoldt, F., Erdt, M.: Window detection in aerial texture images of the berlin 3d citygml model. arXiv preprint arXiv:1812.08095 (2018)
6. Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al.: Swin transformer v2: Scaling up capacity and resolution. arXiv preprint arXiv:2111.09883 (2021)
7. Müller, P., Zeng, G., Wonka, P., Van Gool, L.: Image-based procedural modeling of facades. ACM Trans. Graph. **26**(3), 85 (2007)
8. Ng, P.C., Henikoff, S.: Sift: Predicting amino acid changes that affect protein function. Nucleic acids research **31**(13), 3812–3814 (2003)
9. Nordmark, N., Ayenew, M.: Window detection in facade imagery: A deep learning approach using mask r-cnn. arXiv preprint arXiv:2107.10006 (2021)
10. Osmulski, R.: Yolo open images (2018), *https* : *//github.com/radekosmulski/yolo*$_o$*pen*$_i$*mages*
11. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement (2018)
12. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks (2020)
13. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: Loftr: Detector-free local feature matching with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8922–8931 (2021)
14. Xue, N., Wu, T., Bai, S., Wang, F.D., Xia, G.S., Zhang, L., Torr, P.H.S.: Holistically-attracted wireframe parsing (2020)
15. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. arXiv preprint arXiv:2110.06864 (2021)