FAST AND REGULARIZED RECONSTRUCTION OF BUILDING FAÇADES FROM STREET-VIEW IMAGES USING BINARY INTEGER PROGRAMMING

Han Hu[a, b, *], Libin Wang[b], Mier Zhang[b], Yulin Ding[a, b], Qing Zhu[b]

[a] State Key Laboratory of Rail Transit Engineering Informatization, China Railway First Survey and Design Institute Co. Ltd., Xi'an, China
[b] Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu
(han.hu@swjtu.edu.cn, wlb@my.swjtu.edu.cn, mierzhang@my.swjtu.edu.cn, rainforests@126.com, zhuq66@263.net)

Commission II, WG II/4

KEY WORDS: Façade Reconstruction, YOLOv3, Binary Integer Programming, Regularization, Street-View Images

ABSTRACT:

Regularized arrangement of primitives on building façades to aligned locations and consistent sizes is important towards structured reconstruction of urban environment. Mixed integer linear programing was used to solve the problem, however, it is extreamly time consuming even for state-of-the-art commercial solvers. Aiming to alleviate this issue, we cast the problem into binary integer programming, which omits the requirements for real value parameters and is more efficient to be solved . Firstly, the bounding boxes of the primitives are detected using the YOLOv3 architecture in real-time. Secondly, the coordinates of the upper left corners and the sizes of the bounding boxes are automatically clustered in a binary integer programming optimization, which jointly considers the geometric fitness, regularity and additional constraints; this step does not require *a priori* knowledge, such as the number of clusters or pre-defined grammars. Finally, the regularized bounding boxes can be directly used to guide the façade reconstruction in an interactive envinronment. Experimental evaluations have revealed that the accuracies for the extraction of primitives are above 0.82, which is sufficient for the following 3D reconstruction. The proposed approach only takes about 10% to 20% of the runtime than previous approach and reduces the diversity of the bounding boxes to about 20% to 50%.

1. INTRODUCTION

Reconstruction of building façades is one of the key steps towards complete reconstruction of a LOD-3 (Level-of-Details) model in CityGML protocol (Gröger, Plümer, 2012). Semantic objects such as windows, doors, and balconies are important components of a building façade. Extracting them (Hoegner, Stilla, 2015) and arranging them in a regularized manner (Hensel et al., 2019) are two important steps towards structured LOD-3 reconstruction (Zhu et al., 2020). And the street-view image is arguably the best option for the above objectives due to the public availability and effectiveness in collecting, such as the Google street map (Anguelov et al., 2010).

For the detection of semantic objects in street-view images, classical methods include the use of projected histograms (Lee, Nevatia, 2004, Kostelijk, 2012), gradient projection, K-means clustering (Recky, Leberl, 2010), correlation coefficient (Mayer, Reznik, 2007), perceptual grouping (Sirmacek et al., 2011) and *etc.*. Such methods do not consider the structural and spatial distribution of the semantic objects. Recently, methods based on deep learning (Mathias et al., 2016, Liu et al., 2017) have been widely used to extract the semantic objects on building façade, which have achieved impressive results on images with projective distortion and scale difference; but the regularities of semantic objects have not been considered yet.

In general, these semantic objects should conform to certain regularities, such as aligned locations and consistent sizes. However, due to the characteristics of projection distortion and complex background, the geometric attributes of the extracted primitives in images of buildings façade are generally deviated slightly from the expected. Although the regularization of 2D boundaries, such as edges of buildings, are widely studied in the community (Xie et al., 2018), the approaches cannot be directly adopted. In addition, the regular arrangements of façades can also be learned for specific scenarios (Dehbi, Plümer, 2011, Dehbi et al., 2017); however, the learned models can only be used in inductive fashion, *e.g.* it does not generalize to unseen data.

Recently, a general and promising approach to align different objects of building façades using Mixed Integer Linear Programming (MILP) was proposed (Hensel et al., 2019). However, in our practice the MILP is too complex to solve, which requires prohibitively high runtime consumption. Because we are aiming to integrate the pipeline into an interactive reconstruction environment, at least near real-time response of the solver is required. To solve this issue, we reformulate the problem as a Binary Integer Programming (BIP), with all the unknowns in the binary space of $\{0,\ 1\}$, and the objective can be expressed explicitly as logical operations of the binary variables. Rather than MILP, the BIP is relatively more efficient to be handled by state-of-the-art optimization routines (Gleixner et al., 2018, Gurobi, 2014).

In summary, this paper proposes a fast and regularized reconstruction methods for the façades of buildings from street-view images. Firstly, we extract typical façade primitives using real-time object detection pipeline, *e.g.* the YOLOv3 architectural (Redmon et al., 2016, Redmon, Farhadi, 2018). Secondly, the positions and sizes of the primitives are clustered using BIP by optimizing two competing desires of retaining the best fitness and regularities, for which we require no extra information of the façades. At last, the primitives after clustering are recon-

* Corresponding Author

structed in an interactive environment, *e.g.* SketchUp, by substituting each clustered primitive with a pre-built component model or interactively sketching the component on street-view images.

2. RELATED WORKS

A lot of works have been devoted to extraction and segmentation of building façades, in the communities of photogrammetry, computer vision and computer graphics. With regard to detecting façade objects from images, in recent years, various deep learning architectures, such as CNN (Krizhevsky et al., 2012) and RNN (Graves et al., 2008), have achieved impressive results for various computer vision tasks, such as image classification (Chan et al., 2015) and object detection (Girshick et al., 2015). Although earlier CNN architectures can greatly improve the accuracy of object detection, the detection rate is very slow. This is because that several segregated steps (Girshick et al., 2015) are used, including generation of proposals and classification of the regions. For this reason, the usage in applications requiring real-time responses is limited. The YOLO (You Only Look Once) network (Redmon et al., 2016, Redmon, Farhadi, 2018), as the name suggested, only requires a single integrated forward passing in the testing stage and achieves real-time detection rates for off-the-shelf video sensors. The incrementally upgraded YOLOv3 (Redmon, Farhadi, 2018), due to the integration of ResNet (He et al., 2016), FPN (Feature Pyramid Network) (Lin et al., 2017), and binary cross entropy loss, greatly improves both detection speed and detection accuracy. In the meantime, it has also increased the performance on small targets, which is suitable for detecting semantic objects with complex repeating structures on the building façade. And therefore, this paper adopts the YOLOv3 as the backbone for the detection of the primitives.

With regard to the regular arrangements of objects, based on explicit or implicit procedural methods, the structure of façade was inferred through grammatical rules, including random grammar (Alegre, Dellaert, 2004), syntax trees (Ripperda, Brenner, 2006), and the bottom-up or top-down hybrid approach (Han, Zhu, 2008). They all required setting the correct parameters of the shape syntax in advance. Although these methods have achieved good results, they assume that the image is composed of a fairly regular grid; in addition, fixed expressions of the grammars are not capable to cover the diversities in real-world applications. Procedural grammars are also quite cumbersome to be edited and compiled, which requires tremendous expert knowledge. Human intervention is also required to select the appropriate grammar for a particular building. Although style classifiers (Mathias et al., 2016) was developed to alleviate the above issues, which automatically recognized architectural styles from low-level image features, the use of style syntax is still needed in advance, which is probably a limitation for this approach.

Recent approaches based on mixed integer programming is arguably the most flexible and powerful tool for the problem of regular arrangement of objects. It has been used for arrangements of the 2D boundaries and 3D planes (Monszpart et al., 2015), reconstruction of surface meshes (Boulch et al., 2014, Nan, Wonka, 2017), modeling of the roof structures of the LOD-2 models (Goebbels, Pohle-Fröhlich, 2019) and the façades (Hensel et al., 2019). However, most of them formulated the optimization problem as MILP (Goebbels, Pohle-Fröhlich, 2019, Hensel et al., 2019) or even mixed integer nonlinear programming (Monszpart et al., 2015), which has unknowns in both spaces of integer and real values. Unfortunately, this kind of problems raised up in the operational research has no efficient solvers for large scale problems, even using state-of-the-art commercial libraries (Gurobi, 2014). A practical remedy is to reformulate the problem into BIP (Nan, Wonka, 2017, Kelly et al., 2017, Kelly, Mitra, 2018), which only considers binary variables and linear energies; the regularities can still be explicitly modeled through the logical operations using the binary variables and there are relatively more efficient solvers for these simpler problems. Therefore, we use BIP to model the regularization problem of the façade objects.

3. DETECTION OF FAÇADE PRIMITIVES USING YOLOV3

We use YOLOv3 (Redmon, Farhadi, 2018) to detect axis-aligned bounding boxes of primitives because of its real-time performance. For completeness, we briefly introduce the architecture and implementation details of YOLOv3 here. Rather than other region-based CNN methods (Girshick et al., 2015), YOLO (Redmon et al., 2016) uses regression to directly process the entire image, and obtains categories and positions of the targets in a single forward propagation. YOLO implements an end-to-end pipeline for detection by dividing the image into $s \times s$ grids. If the center of the semantic component is in a grid, the grid is responsible for predicting the target. Each grid will generate B bounding boxes, and each bounding box must predict its confidence χ, which is defined as the product of the probability P of the target contained in the bounding box and the accuracy Q, as $\chi = P \times Q$. If the grid contains semantic objects, then $P = 1$, otherwise $P = 0$. Q represents the intersection ratio of the labeled box in training samples and the predicted box. When $Q = 1$, it means that the labeled box and the predicted box coincide perfectly.

If a grid contains semantic components, which corresponds to C classes, it is represented by P_i for each class. Therefore, we can obtain the intermediate score of each grid and each class as $\phi_i = P_i \times \chi$. The scores are truncated at 0 and non-maximum suppression is used to remove bounding boxes with a large repetition rate. In the end, each bounding box only retains the objects with positive confidence scores and the highest categories. In YOLOv3, in order to improve the accuracy of target detection, the residual network (He et al., 2016) is used as backbone. The features before entering the residual box and the features output by the residual box are combined to extract deeper feature information. On the building façade, even if they are the same type of semantic objects, their sizes and poses are not the same. YOLOv3 uses multi-scale fusion (Lin et al., 2017) to detect objects, and has good adaptability to the scale changes of objects.

4. REGULAR ARRANGEMENTS OF FAÇADE PRIMITIVES USING BINARY INTEGER PROGRAMMING

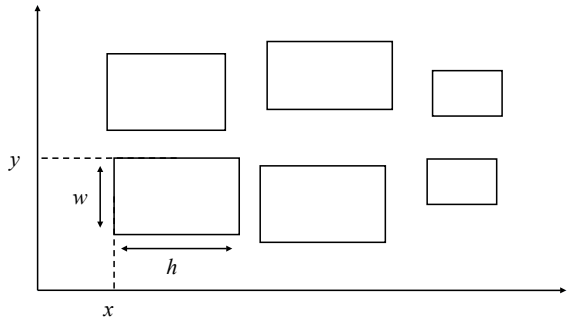
After initial extraction of the bounding boxes of the building façade, we then use BIP to restore the spatial regularity of the windows, doors and balconies, inspired by previous work (Hensel et al., 2019). Although the MILP method has been successfully used in many studies (Boulch et al., 2014, Hensel et al., 2019), in our pipeline, because we are aiming at an interactive reconstruction pipeline, the runtime should be kept reason-

ably low. In the following, we describe our reformulated problem setup using BIP instead of MILP.

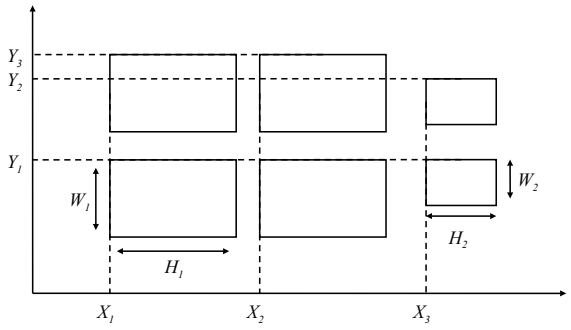
4.1 Problem setup using binary integer programming

After extracting the initial primitives, we have N bounding boxes for each image, and each bounding box is uniquely determined by a set of four parameters (x, y, w, h) , where (x, y) and (w, h) are coordinate of the upper left corner and size of the bounding box, respectively (Figure 1a). Instead of directly optimizing these parameters that are real values using MILP (Hensel et al., 2019), we cast it into a model selection problem using BIP.

Specifically, we first establish a model space for each *attribute* of the bounding box, *e.g.* $\mathbf{X} = \{X_1, X_2, ..., X_N\}$ for the attribute of x coordinate. The size of $|\mathbf{X}|$ could be the number of bounding boxes N, but we choose to compress it by pre-cluster the model space using a very confident lower bound δ_l as described later. We then assign a binary variable $a^x_{i,k} \in \{0, 1\}$ to represent the state of the selection, *i.e.* if the model X_k is selected for the attribute x of the i_{th} bounding box. In addition, we use the one-hot vector $\boldsymbol{\xi}_i$ [1] to represent the whole state of the i_{th} bounding box as $\boldsymbol{\xi}_i = (a_{i,0}, a_{i,1}, ..., a_{i,|\mathbf{X}|})^T$.



(a) Parameters for primitives



(b) Parameters for model spaces

Figure 1. Parameters for primitives and model spaces.

In fact, the model spaces of the attributes of the primitives for a single façade are generally quite limited in urban environment. That is the ratio $N/|\mathbf{X}|$ is generally quite large, which leads to unnecessarily too many parameters. Therefore, we pre-cluster all the attributes separately using the mean shift approach (Cheng, 1995); and the threshold is set to the lower bound δ_l. The values in the model space $\mathbf{X}$ are determined by the centers of the clusters, as shown in Figure 1b. To ensure the accuracies of the results, the lower bound δ_l in mean shift clustering should be as small as possible to avoid aggregating parameters of different properties into the same category. It should be noted that, although the same threshold δ_l is used for all the attributes, the number of clusters $|\mathbf{X}|$, $|\mathbf{Y}|$, $|\mathbf{W}|$ and $|\mathbf{H}|$ are generally different.

In summary, the purpose is to optimize all the selecting vectors $\boldsymbol{\xi}$, under the energy functions and constraints as described below. And the total size of explicit unknowns is $N \times (|\mathbf{X}| + |\mathbf{Y}| + |\mathbf{W}| + |\mathbf{H}|)$.

4.2 Energy functions to be optimized

Our loss function consists of a data item and a regularity item. First of all, our goal is to make the sum of the changes of the bounding boxes against the initial locations as small as possible after the regularization. Therefore, we first calculate the residual vector $\boldsymbol{\epsilon}$ for each bounding box, which represents the errors for different selections, as

$$\boldsymbol{\epsilon}^x_i = (x_i - X_0, x_i - X_1, ..., x_i - X_{|X|})^T, \tag{1}$$

where the superscript x denotes different attributes.

In this way, the total energy O^a_d for attribute a caused by the selection vectors, *e.g.* offsets for the coordinates of upper left corners and differences for the sizes of the rectangles, can be briefly expressed as,

$$O^a_d = \sum_i^N |\boldsymbol{\epsilon}^a_i| \cdot \boldsymbol{\xi}^a_i. \tag{2}$$

Equation 2 means that, for each bounding box, we only account for the error of the selected value in model space, *i.e.* when $a_{i,k} = 1$. The final data term of the energy function is therefore intuitively the summation of all the attributes as

$$O_d = O^x_d + O^y_d + O^w_d + O^h_d. \tag{3}$$

With only the data term, we always have a trivial solution that have the best fit, *e.g.* choosing the nearest center of the mean shift clustering. Therefore, we introduce a regularity item. The intuition behind this term is that higher regularity generally means less categories; fortunately, the number of selected categories is easy to model as illustrated in Figure 2. For each attribute a, the total number of selected categories, *e.g.* the regularity term O^a_g, can be explicitly expressed as the following logical expression,

$$O^a_g = \|\boldsymbol{\xi}^a_1 \vee \boldsymbol{\xi}^a_2 \vee ... \vee \boldsymbol{\xi}^a_N\|_1, \tag{4}$$

where $\|\cdot\|_1$ is the L_1 norm that is the absolute summation of all the elements of a vector and for binary variables L_1 norm simply counts the number of non-zero variables; the binary operator $\vee$ is the element-wise *logical or* for the one-hot vectors. Similar to Equation 3, the final regularity term is a weighted summation across all the attributes as

$$O_g = \omega^x O^x_g + \omega^y O^y_g + \omega^w O^w_g + \omega^h O^h_g, \tag{5}$$

where ω denotes the weights of different attributes. And the final energy function is

$$O = O_d + O_g. \tag{6}$$

[1] We omit the superscript for attribute when not ambiguous. In addition the Greek symbols are used for one-hot vectors and Roman symbols for variables.

$$\begin{matrix} \xi_1 & \xi_2 & \xi_3 & \xi_4 & \xi_5 \end{matrix} \quad or \quad L_1$$

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = 3$$

Figure 2. Illustration of the regularity energy as in Equation 4, which is the total number of selected models.

4.3 Constraints of the binary integer programming

The variables $\boldsymbol{\xi}_i = (a_{i,0}, a_{i,1}, ..., a_{i,N})^T$ cannot be adjusted freely. Obviously, because each bounding box can only choose one state, we have the following constraint C_1 for each bounding box,

$$C_1 : \sum_k a_{i,k} = 1, \forall\, i \leq N. \tag{7}$$

Another practical constraint c_2 is that we could very confidently ignore certain model spaces if the residual $|\boldsymbol{\epsilon}_{i,k}|$ exceeds an upper bound δ_u, as.

$$C_2 : a_{i,k} = 0,\ \forall\, |\boldsymbol{\epsilon}_{i,k}| > \delta_u. \tag{8}$$

It seems the additional constraints may increase the complexity of the problem, but interestingly, in practice, we find that the additional constraints significantly reduce the runtime, with almost no differences in the final results.

4.4 Implementation details

The implementation of Equation 4 needs some tricks, because it involves the logical operations. For two binary variable a and b, the *logical or* result $c = a \vee b$ can be modeled by adding the following constraints,

$$\begin{aligned} c \leq & a + b \\ c \geq & a \\ c \geq & b \end{aligned} \quad . \tag{9}$$

In fact, this kind of fixed routines can be handled efficiently and gracefully by state-of-the-art solvers (Chinneck, 2007, Gurobi, 2014). For the parameters, we set $\delta_l \in [3, 5]$ pixels and $\delta_u = 10\delta_l$; and $\omega^x = \omega^y = 100$ and $\omega^w = \omega^h = 10$ are used empirically. In this way, all the energy functions and constraints are linear functions, which are solved using the Mosek library (Mosek, 2010).

5. EXPERIMENTAL EVALUATIONS

5.1 Evaluation of detections of façade primitives

This paper uses the CMP façade database (Tylecek, 2012) as the training data set, which contains a total of 606 building façade images around the world. These images are manually labeled with 12 semantic objects on the façade. We choose three typical primitives: window, door and balcony. We built the YOLOv3 model based on Keras (Gulli, Pal, 2017) to train the above data set. At the same time, we took 30 typical building façade images from Google street view (Anguelov et al., 2010) for testing, and manually labeled them for evaluations. In order to verify the effectiveness of this method, we adopted the same evaluation method in (Rahmani, Mayer, 2018). We counted every classified pixel as either true positive(TP) or false positive(FP), and the precision is thus $TP/(TP+FP)$. On our test dataset, for windows, doors, and balconies, our average extraction precision reached 0.917, 0.856, and 0.852.

In addition, we used the same test dataset ICG Graz50 (Riemenschneider et al., 2012) to compare with a recent method (Hensel et al., 2019) based on Fatser RCNN (Table 1). Both methods are trained on the CMP dataset. The precisions of the extraction of windows and doors listed in (Hensel et al., 2019) are 0.892 and 0.834, and the precision of our method based on YOLOv3 are 0.882 and 0.825. Considering that YOLOv3 is more efficient than Faster RCNN, the mild precision loss is acceptable. And the detection performance could be considered satisfactory.

Table 1. Comparison of precisions on the ICG Graz50 dataset, with model trained on the CMP façade dataset.

	Window	Door
Hensel et al. (2019)	0.892	0.834
Proposed	0.882	0.825

We tested the precision of window extraction using images with different resolutions and different layout complexity (Table 2). The scale is measured by downsampling the images and the layout complexity is the total number of the selected modes on the of the four attributes of the windows. It can be seen from Table 2 that when the resolution of the image is within a certain range, the extraction precision is good and the difference is small; but when the resolution is too low, the extraction precision is significantly reduced. In addition, it can be noticed that as the layout complexity increases, the extraction precision tends to gradually decrease. In summary, when the images are captured at a relatively high resolution, the layout complexity has a greater impact on the extraction precision.

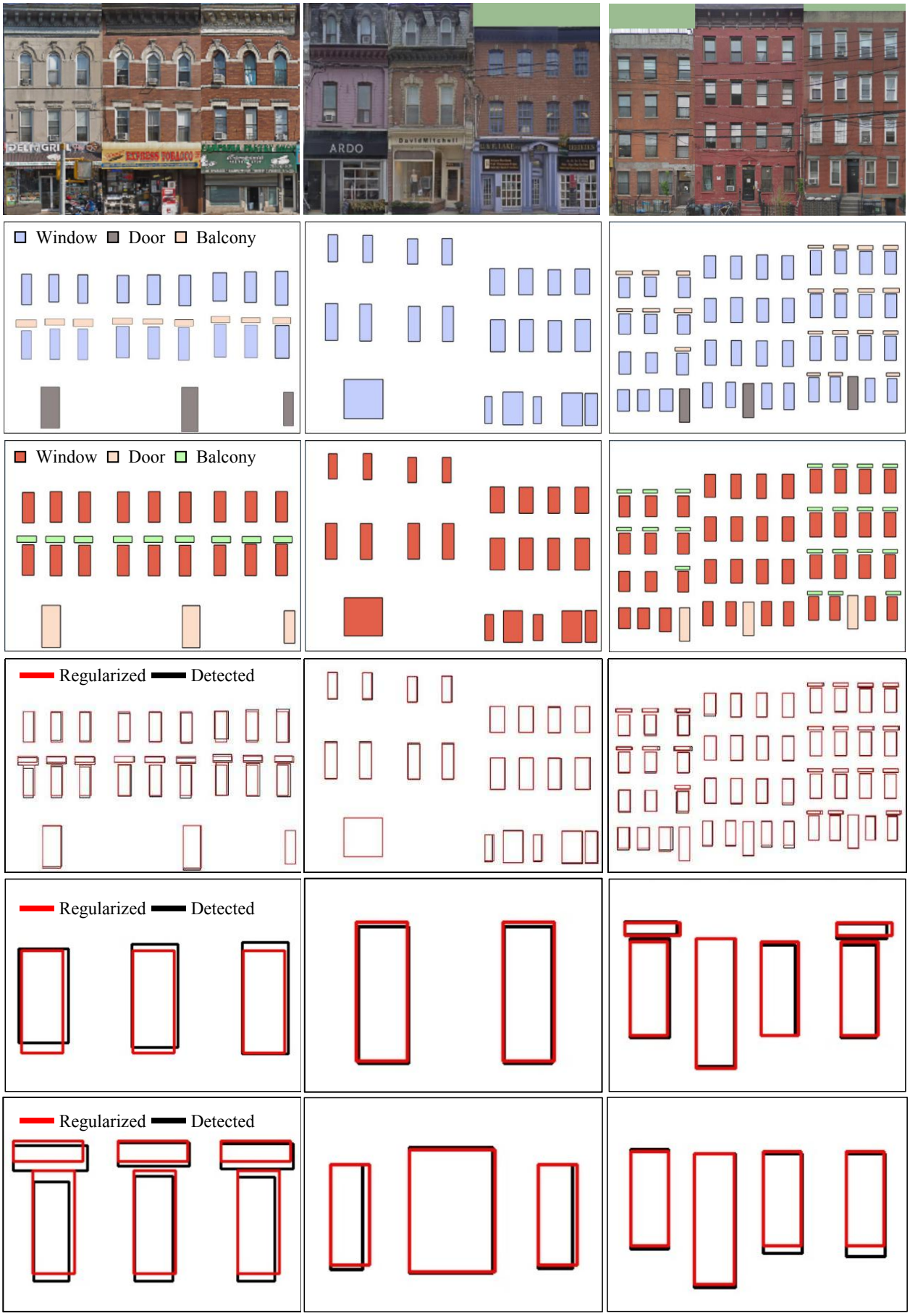
Table 2. Detection performance with respect to different scale of images and different complexities of façade layout.

Scale	1	1/2	1/4	1/8	1/16
Precision	0.854	0.845	0.893	0.895	0.565
Complexity	32	36	56	76	92
Precision	0.928	0.916	0.908	0.868	0.82

5.2 Evaluation and comparisons of the regular arrangements of the primitives

We selected three typical building façade images of three cities in the United States (US), United Kingdom (UK), and Canada (CA) to evaluate the performance of the regularization. Both qualitative and quantitative evaluations are conducted and we also compare the runtime performance against the MILP approach (Hensel et al., 2019).

Qualitative evaluations. Figure 3 compares the extracted and regularized bounding boxes for the US, UK and CA datasets. The black frame represents the extracted primitives and the red frame indicates the regularized results. It can be noticed that after regularization, the semantic objects on the building



(a) CA (b) UK (c) US

Figure 3. Comparison of detected and regularized façade primitives for the three datasets. The second and third rows show the detected and regularized primitives, respectively. The fourth row compares them and the last two rows give two enlarged demonstrations.

façade are arranged more neatly and consistently and still fit well enough to the original bounding boxes, as demonstrated in the enlarged regions. In addition, Figure 4 demonstrates the reconstructed façades for the three datasets in off-the-shelf modeling solutions.

(a) US

(b) UK

(c) CA

Figure 4. Reconstructed façades for the three datasets, in which the right column demonstrates the enlarged areas in the cyan rectangles.

Quantitative evaluations. We counted the number of used model space before and after the regularization to measure the regularity of the results, e.g. O_g^a in Eq. 4. Table 3 demonstrates the results, and it could be noted that, the selected parameters only account for about 50% for the coordinates of the corners and 20% for the sizes.

Table 3. Quantitative evaluations of the regularity by the size of the model space before and after regularization, *i.e.* O_g^a in Eq. 4. The number of the selected model space significantly reduced after optimization.

Dataset	#Detected				#Regularized			
	$\|\mathbf{X}\|$	$\|\mathbf{Y}\|$	$\|\mathbf{W}\|$	$\|\mathbf{H}\|$	$\|\mathbf{X}\|$	$\|\mathbf{Y}\|$	$\|\mathbf{W}\|$	$\|\mathbf{H}\|$
US	76	62	35	39	38	25	5	4
UK	22	20	17	17	16	6	6	6
CA	47	39	29	26	31	6	10	5

Comparisons of runtime. In order to verify the efficiency of the method in this paper, we tested six building façades with complex structures and numerous parameters, and compared the proposed BIP approach against the MILP approach (Hensel et al., 2019). The results are shown in the Table 4 and the runtime of the proposed BIP approach only accounts for about 10% to 20% of the MILP approach. For the MILP approach (Hensel et al., 2019), the number of explicit unknown parameters are $N(2|\mathbf{X}| + 2|\mathbf{Y}| + |\mathbf{W}| + |\mathbf{H}|) + 8N$, including $8N$ real value parameters. In the proposed approach, the number of explicit unknown parameters is $N(|\mathbf{X}| + |\mathbf{Y}| + |\mathbf{W}| + |\mathbf{H}|)$. Although the proposed method has slightly fewer parameters, the numbers are still in the same order of magnitude. Therefore, it is the reformulated problem that account for the performance differences.

Table 4. Comparison of the runtime between MILP and the proposed BIP approaches. The second to fifth columns demonstrates the complexities of the model space.

N	$\|\mathbf{X}\|$	$\|\mathbf{Y}\|$	$\|\mathbf{W}\|$	$\|\mathbf{H}\|$	MILP (s)	BIP (s)
26	11	5	3	2	5.7	0.9
74	20	13	3	3	150.9	19.9
60	29	10	4	7	135.2	20.8
61	10	16	4	7	84.6	12.7
67	24	6	4	5	106.2	16.6
45	35	12	9	9	123.6	20.3

6. CONCLUSION

This paper proposed an approach for the regular arrangement of primitives of the building façades using BIP. Compared to the MILP approach, BIP is considerably faster and achieves near real-time performance with similar level of data fitness and regularities. The detected and rearranged bounding boxes of the primitives can be directly used for the modeling of the façade features, which is a key step towards the LOD-3 reconstruction. However, current approaches can only detect axis-aligned objects, future works may be devoted to explore the reconstruction of more complex façade features. Code is available at https://github.com/saedrna/Ranger.

ACKOWLEDGEMENT

This paper is supported by the National Natural Science Foundation of China (Project No.: 41631174, 61602392, 41871291).

REFERENCES

Alegre, F., Dellaert, F., 2004. A probabilistic approach to the semantic interpretation of building façades.

Anguelov, D., Dulong, C., Filip, D., Frueh, C., Lafon, S., Lyon, R., Ogale, A., Vincent, L., Weaver, J., 2010. Google street view: Capturing the world at street level. *Computer*, 43(6), 32–38.

Boulch, A., de La Gorce, M., Marlet, R., 2014. Piecewise-planar 3D reconstruction with edge and corner regularization. *Computer Graphics Forum*, 33(5), 55–64.

Chan, T.-H., Jia, K., Gao, S., Lu, J., Zeng, Z., Ma, Y., 2015. PCANet: A simple deep learning baseline for image classification. *IEEE Transactions on Image Processing*, 24(12), 5017–5032.

Cheng, Y., 1995. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8), 790–799.

Chinneck, J. W., 2007. *Feasibility and Infeasibility in Optimization:: Algorithms and Computational Methods*. 118, Springer Science & Business Media.

Dehbi, Y., Hadiji, F., Gröger, G., Kersting, K., Plümer, L., 2017. Statistical relational learning of grammar rules for 3D building reconstruction. *Transactions in GIS*, 21(1), 134–150.

Dehbi, Y., Plümer, L., 2011. Learning grammar rules of building parts from precise models and noisy observations. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(2), 166–176.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2015. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1), 142–158.

Gleixner, A., Bastubbe, M., Eifler, L., Gally, T., Gamrath, G., Gottwald, R. L., Hendel, G., Hojny, C., Koch, T., Lübbecke, M. E., Maher, S. J., Miltenberger, M., Müller, B., Pfetsch, M. E., Puchert, C., Rehfeldt, D., Schlösser, F., Schubert, C., Serrano, F., Shinano, Y., Viernickel, J. M., Walter, M., Wegscheider, F., Witt, J. T., Witzig, J., 2018. The scip optimization suite 6.0. Technical report, Optimization Online.

Goebbels, S., Pohle-Fröhlich, R., 2019. Beautification of city models based on mixed integer linear programming. *Operations Research Proceedings 2018*, Springer, 119–125.

Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J., 2008. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5), 855–868.

Gröger, G., Plümer, L., 2012. CityGML–Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71, 12–33.

Gulli, A., Pal, S., 2017. *Deep learning with Keras*. Packt Publishing Ltd.

Gurobi, 2014. Gurobi optimizer reference manual, version 6.0. http://www.gurobi.com.

Han, F., Zhu, S.-C., 2008. Bottom-up/top-down image parsing with attribute grammar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1), 59–73.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *CVPR*, 770–778.

Hensel, S., Goebbels, S., Kada, M., 2019. Façade reconstruction for textured LOD2 citygml models based on deep learning and mixed integer linear programming. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4.

Hoegner, L., Stilla, U., 2015. Building façade object detection from terrestrial thermal infrared image sequences combining different views. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(3), 55.

Kelly, T., Femiani, J., Wonka, P., Mitra, N. J., 2017. BigSUR: large-scale structured urban reconstruction. *ACM Transactions on Graphics (TOG)*, 36(6), 204.

Kelly, T., Mitra, N. J., 2018. Simplifying Urban Data Fusion with BigSUR. *arXiv preprint arXiv:1807.00687*.

Kostelijk, T., 2012. Semantic annotation of urban scenes: Skyline and window detection. PhD thesis, Doctoral dissertation, Universiteit van Amsterdam.

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1097–1105.

Lee, S. C., Nevatia, R., 2004. Extraction and integration of window in a 3d building model from ground view images. *CVPR*.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. *CVPR*, 2117–2125.

Liu, H., Zhang, J., Zhu, J., Hoi, S. C., 2017. DeepFacade: A deep learning approach to façade parsing.

Mathias, M., Martinović, A., Van Gool, L., 2016. ATLAS: A three-layered approach to façade parsing. *International Journal of Computer Vision*, 118(1), 22–48.

Mayer, H., Reznik, S., 2007. Building façade interpretation from uncalibrated wide-baseline image sequences. *ISPRS Journal of Photogrammetry and Remote Sensing*, 61(6), 371–380.

Monszpart, A., Mellado, N., Brostow, G. J., Mitra, N. J., 2015. RAPter: rebuilding man-made scenes with regular arrangements of planes. *ACM Transactions on Graphics*, 34(4), 103–1.

Mosek, A., 2010. The MOSEK optimization software. 54(2-1), 5. Online at http://www.mosek.com.

Nan, L., Wonka, P., 2017. Polyfit: Polygonal surface reconstruction from point clouds. *ICCV*, 2353–2361.

Rahmani, K., Mayer, H., 2018. High quality façade segmentation based on structured random forest, region proposal network and rectangular fitting. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV/II.

Recky, M., Leberl, F., 2010. Windows detection using k-means in cie-lab color space. *2010 20th International Conference on Pattern Recognition*, IEEE, 356–359.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. *cvpr*, 779–788.

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Riemenschneider, H., Krispel, U., Thaller, W., Donoser, M., Havemann, S., Fellner, D., Bischof, H., 2012. Irregular lattices for complex shape grammar facade parsing. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 1640–1647.

Ripperda, N., Brenner, C., 2006. Reconstruction of façade structures using a formal grammar and rjmcmc. *Joint Pattern Recognition Symposium*, Springer, 750–759.

Sirmacek, B., Hoegner, L., Stilla, U., 2011. Detection of windows and doors from thermal images by grouping geometrical features. *2011 Joint Urban Remote Sensing Event*, IEEE, 133–136.

Tylecek, R., 2012. The cmp façade database. Technical report, Tech. rep., CTU–CMP–2012–24, Czech Technical University.

Xie, L., Zhu, Q., Hu, H., Wu, B., Li, Y., Zhang, Y., Zhong, R., 2018. Hierarchical Regularization of Building Boundaries in Noisy Aerial Laser Scanning and Photogrammetric Point Clouds. *Remote Sensing*, 10(12), 1996.

Zhu, Q., Zhang, M., Hu, H., Wang, F., 2020. Interactive correction of a distorted street-view panorama for efficient 3d façade modeling. *IEEE Geoscience and Remote Sensing Letters*, 1-5. 10.1109/LGRS.2019.2962696.