Adaptive Partitioning of Urban Facades

Chao-Hui Shen[1] Shi-Sheng Huang[1] Hongbo Fu[2] Shi-Min Hu[1]
[1]TNList, Tsinghua University, Beijing [2]City University of Hong Kong

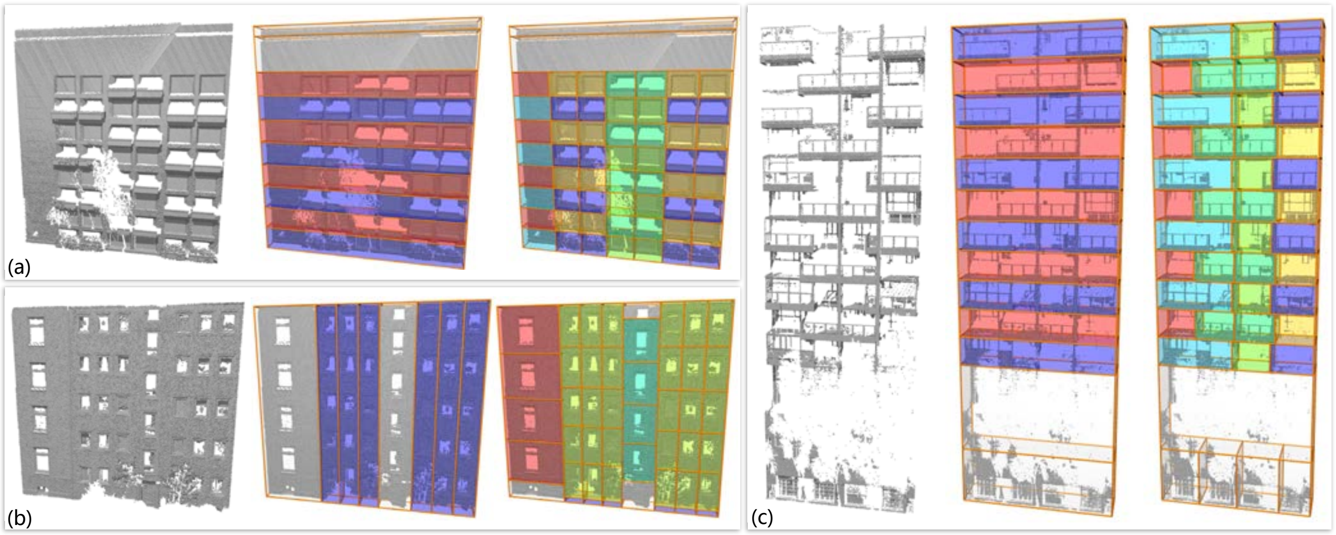


Figure 1: *Given input unorganized point clouds of 3D urban facades, a recursive adaptive partitioning is automatically performed to build a hierarchy of building blocks upon them (from left to right). The splitting direction, number and location of splitting planes are all adaptively determined in each step. Repetitive patterns are indicated by different colors.*

Abstract

Automatically discovering high-level facade structures in unorganized 3D point clouds of urban scenes is crucial for applications like digitalization of real cities. However, this problem is challenging due to poor-quality input data, contaminated with severe missing areas, noise and outliers. This work introduces the concept of *adaptive partitioning* to automatically derive a flexible and hierarchical representation of 3D urban facades. Our key observation is that urban facades are largely governed by concatenated and/or interlaced grids. Hence, unlike previous automatic facade analysis works which are typically restricted to globally rectilinear grids, we propose to automatically partition the facade in an adaptive manner, in which the splitting direction, the number and location of splitting planes are all adaptively determined. Such an adaptive partition operation is performed recursively to generate a hierarchical representation of the facade. We show that the concept of adaptive partitioning is also applicable to flexible and robust analysis of image facades. We evaluate our method on a dozen of LiDAR scans of various complexity and styles, and the image facades from the eTRIMS database and the Ecole Centrale Paris database. A series of applications that benefit from our approach are also demonstrated.

Links: DL PDF WEB

1 Introduction

With the recent advances in LiDAR scanning devices, the acquisition of 3D point clouds from urban buildings is getting more efficient and more convenient. However, the captured point cloud often suffers from severe missing data, noise and outliers, making the reconstruction of architectural models with faithful geometry and topology from such data rather challenging. The key to this problem is to explore and utilize the characteristics of urban scenes as prior knowledge, especially repetition of building elements in facades. To obtain such architectural characteristics, the state-of-the-art works [Zheng et al. 2010; Nan et al. 2010] rely on user assistance, which would be labor intensive for applications like digitalization of real cities.

There exist automatic solutions for discovering facade structures in single- or multi-view facade images [Müller et al. 2007; Xiao et al. 2009; Musialski et al. 2010]. A common assumption made in those works is that facades are inherently governed by *global rectilinear structures*. That is, a facade can be split into building blocks (e.g., windows) by a single rectilinear grid. Although there indeed exist many real facades satisfying such assumption, it is not general enough to handle many other patterns like asymmetric patterns (Figure 1), which are also ubiquitous in urban scenes.

This work aims for a more flexible representation of high-level facade structures, which is based on the key observation: the high-level structure of a facade is largely governed by either a rectilinear grid or a mixture of rectilinear grids by concatenation and/or interlacing. For examples, Figure 2 shows facades with concatenated grids and interlaced grids. Therefore, how to identify different grids of repetitive elements and their relations is our core problem.

To address the problems, the concept of *adaptive partitioning* of urban facades is introduced in this paper (Figure 1). Explicitly searching for repetitive patterns over the entire facade is challenging given poor data quality and complex hidden structures. The underlying

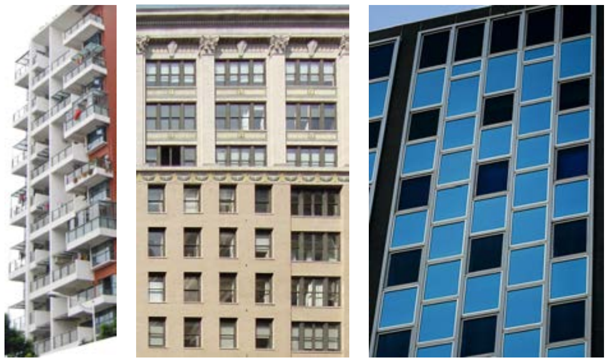


Figure 2: *Facades with concatenated grids and/or interlaced grids of architectural elements are ubiquitous in urban scenes.*

rectilinear mixture model largely implies that facade elements are often globally aligned along at least one direction (i.e., either horizontal or vertical) but not necessarily both horizontal and vertical directions. Therefore, we propose to automatically partition the facade into multiple sub-facades in an adaptive manner, each of which is possibly governed by a rectilinear grid of architectural elements and thus then grouped by similarity for further partitioning. Applying such partition recursively leads to a hierarchical representation of the facade, which benefits various applications such as facade consolidation, manipulation and admixture (Section 4.2).

1.1 Overview

As summarized in Figure 3, each partition step consists of three phases: splitting, grouping and rectification. It can be carried out recursively to generate a top-down hierarchical subdivision of the facade. The same pipeline is applicable to both 3D facade scans and image facades. For brevity, we first focus our discussions on the case of unorganized 3D point clouds as input and discuss its extension to the image domain only in Section 4.3.

Structure Splitting. This step begins with the whole piece of input point cloud and detects horizontal or vertical splitting planes perpendicular to the facade surface, which divide the facade to a list of sub-facades (Section 3.2). The splitting strategy is mainly based on weak prior knowledge of urban buildings, formulated as *penalty functions* of placing horizontal and vertical splitting planes. Each splitting step tackles one direction only (i.e., either horizontal or vertical), which is adaptively chosen by comparing the penalty costs with respect to horizontal and vertical splitting. We also use the penalty functions to adaptively determine the initial number and location of splitting planes, which are later rectified in the rectification step.

Element Grouping. The goal of this step is to group the set of sub-facades resulted from the previous step by similarity (Section 3.3). The grouping of repetitive elements is necessary not only for future applications like non-local consolidation but also for further partition in a consistent manner. The grouping is achieved by geometric registration, followed by iterative bottom-up clustering of pairs of

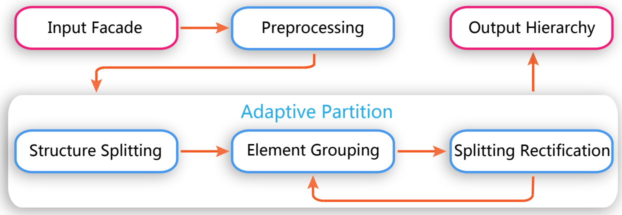


Figure 3: *System overview.*

sub-facades.

Splitting Rectification. The third step is to rectify the number and location of splitting planes by using the grouping information obtained in the second step (Section 3.4). This is important since the initial splitting planes are determined using local strategies and might result in imperfect boundaries of repetitive patterns or miss certain splitting planes in noisy regions. We use the grouping information to first recover the missing splitting planes and then perform a global optimization toward optimal alignment of repetitive facade elements in a least-squares sense. Since to some extent the result of the grouping step is dependent on the splitting position, the grouping and rectification steps are then iterated until convergence.

2 Related Work

There is a large amount of literature on facade, building and architectural analysis and modeling. The review here mainly focuses on existing works involving in-depth analysis of facade structures in urban images/models and discovery of repetitive/symmetry patterns in general 3D models. Reviewing shape analysis works of general point-sampled surfaces [Yamazaki et al. 2010; Bucksch et al. 2010] or modeling-oriented works like procedural facade modeling [Wonka et al. 2003; Müller et al. 2006] is beyond the scope of this paper.

Following early surface reconstruction works like [Levoy et al. 2000], reconstructing urban scenes from terrestrial LiDAR scanning data is an emerging technology in the area of point cloud processing. An automatic data-driven facade reconstruction by cell decomposition is introduced in [Becker and Haala 2009], which, however, requires an additional coarse 3D building model as input. Their algorithm focuses on flat facades containing only windows of shadow depth and thus cannot handle facades with complex geometry (e.g., of balconies). The work of [Ning et al. 2010] also aims for the recovery of a coarse architectural model. To recover more complicated and detailed geometry of buildings, very recent works [Nan et al. 2010; Zheng et al. 2010] require a moderate amount of user intervention to discover underlying architectural structures as repetitive patterns. As an automatic preprocessing tool, our work is complementary to those works.

A few techniques have been specifically designed for discovering facade structures in single images [Müller et al. 2007; Musialski et al. 2010] or multi-view images [Xiao et al. 2008; Xiao et al. 2009]. Such techniques consider the unique characteristics of facade structures such as regularity and orthogonality, and thus often outperform general methods for analysis and detection of repetitive patterns in terms of robustness. However, those techniques are largely based on a strong assumption that a facade is governed by a hidden global rectilinear grid. For example, based on such assumption, Müller et al. [2007] encode the symmetry of a facade by forming an irreducible facade, where the splitting of different repetitive patterns is searched for, always first along the horizontal direction and then the vertical direction. The works of [Xiao et al. 2008; Xiao et al. 2009] decompose the facade along detected image edges, which might lead to an over-segmented partition especially for facades with for example balconies. In addition, the poorer quality of captured geometric data (e.g., much lower resolution, reduced precision or higher level of noise) and additional dimension of data (e.g., due to balconies) make the extension of such image-based methods to 3D facade point clouds challenging.

Our problem is relevant to semantic segmentation of facade images in computer vision (see [Teboul et al. 2011] and the references therein), which aims to not only divide a facade into facade elements (our main goal) but also assign each of them a particular semantic label (e.g., walls, windows, roofs etc). To achieve such semantic parsing, the state-of-the-art works [Teboul et al. 2011; Teboul et al. 2010] heavily rely on supervised multi-class classification, combined with prescribed shape grammars. In contrast, our focus is on facade partitioning of both facade 3D scans and images, with target applications like facade modeling and manipulating. Our technique is model-free, completely unsupervised and simpler to implement.

Our work is also related to regularity or symmetry detection for general 3D models. Most of early works regard the problem as voting for dominant transformations in the space of pairwise similarity transformations [Mitra et al. 2006; Pauly et al. 2008], making isolating interlaced symmetries challenging especially when the number of symmetry groups grows. Based on spatial coherence of symmetry patterns, Bokeloh et al. [2009] present a symmetry detection algorithm that can find rigid symmetries in more general configurations, even for completely irregular patterns of symmetry. Instead of defining suitable similarity measure and matching similar elements directly, as done in those works, our technique intends to find splitting boundaries between similar elements, patterns, and even non-repetitive components, which to some extent is in a spirit of the work of inverse procedural modeling [Bokeloh et al. 2010].

3 Adaptive Partition of Urban Facades

The input to our framework is noisy and incomplete scans of urban buildings acquired by the state-of-the-art terrestrial LiDAR devices, represented as unorganized point clouds. We assume that the point clouds have already been segmented into facades, which will be the focus of our partition algorithm. By construction facades have a dominant planar structure, with the depth variation on the plane. To ease the discussion in this section, we consider the input point cloud of a facade to have a *flat* dominant plane. Extending to general developable surfaces (e.g., cones, cylinders) is trivial given the dominant planar structure detected (Figure 8). Below we first discuss one partition step consisting of three stages (namely, splitting, grouping and rectification), which is performed recursively to form a top-down hierarchical subdivision of the facade.

3.1 Preprocessing

An input facade is first rectified and aligned with three orthogonal axes in the Cartesian coordinate system, corresponding to the ground plane and two additional orthogonal axes [Nan et al. 2010]. The ground up-vector is aligned to the Z-axis, with the front direction to the X-axis and the horizontal direction to the Y-axis.

Next we explicitly detect plane pieces in the original unorganized point cloud using a RANSAC approach [Schnabel et al. 2007] and boundary points [Zheng et al. 2010], which will be shortly used to define penalty functions (Section 3.2). Every boundary point is assigned a direction that approximates the tangent of its associated underlying boundary line. Let λ_2 and λ_1 be the largest and second largest eigen-values of the covariance matrix of the local neighborhood of boundary point $\boldsymbol{p}_i$ (10-nearest *boundary* points used in our implementation), respectively. The direction $\boldsymbol{\mu}_i$ at $\boldsymbol{p}_i$ is then set to be the eigen-vector corresponding to λ_2. A confidence value indicating the likelihood that a point lies along a true boundary is also assigned to $\boldsymbol{p}_i$: $conf(\boldsymbol{p}_i) = 1 - \frac{\lambda_1}{\lambda_2}$. The confidence values lie in the range of [0, 1], with $conf(\boldsymbol{p}_i) = 1$ indicating perfect line distribution. Figure 4 shows an example of detected boundary points, colored by confidence.

3.2 Structure Splitting

In this stage we aim at splitting the current building block into a set of sub-building blocks. For example, when starting from the whole facade as the current block, the facade is expected to be partitioned

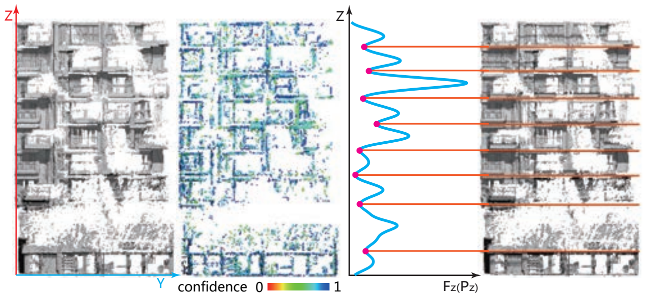


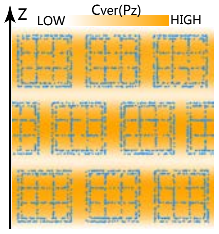
Figure 4: *From left to right: input facade, detected boundary points with confidence, penalty function for horizontal splitting, identified initial horizontal splitting planes.*

into a set of horizontal or vertical slices, depending on which splitting direction is used. The splitting direction, initial number and location of splitting planes are adaptively determined based on some weak prior knowledge of urban buildings, originally observed by Müller et al. [2007]: *horizontal (vertical) splitting planes should be placed where vertical (horizontal) lines are rare and horizontal (vertical) lines are dense.*

To incorporate such prior knowledge into our approach, the degree to which a splitting plane is preferred should be evaluated. Instead of simply counting the number of intersections between a possible splitting plane and the detected boundaries, which is typically fragile, we measure the accumulated effect of vertical/horizontal boundary lines to a certain splitting plane P in a more continuous manner. Taking vertical boundary lines for example, it is formulated as follows:

$$C_{ver}(P) = \sum_{i \in \boldsymbol{B}} \frac{conf(\boldsymbol{p}_i) \cdot \phi(d(\boldsymbol{p}_i, P)) \cdot \|\langle \boldsymbol{\mu}_i, \boldsymbol{v}_{ver} \rangle\|^2}{\rho(\boldsymbol{p}_i)}, \qquad (1)$$

where $\boldsymbol{B}$ is the index set of the detected boundary points within the current building block, $\boldsymbol{v}_{ver} = (0, 0, 1)$ refers to the vertical direction, and the plane P is either horizontal or vertical. The inner product between the direction $\boldsymbol{\mu}_i$ of every boundary point and $\boldsymbol{v}_{ver}$ measures the contribution of $\boldsymbol{\mu}_i$ along the vertical direction, which is weighted by the Gaussian kernel $\phi(d) = e^{-d^2/2\sigma^2}$, with $d(\boldsymbol{p}_i, P)$ defined as the Euclidean distance of a boundary point $\boldsymbol{p}_i$ to the plane P. The parameter σ is related to the size of architectural tiles ($\sigma \equiv 0.5m$ in all our examples). The confidence of the boundary points is also included. $\rho(\boldsymbol{p}_i) = k/V(k)$ is the local point density around $\boldsymbol{p}_i$, where $V(k)$ is the volume of the bounding ball of its k-nearest points in the original point cloud ($k = 10$ used in our implementation, since our LiDAR data roughly samples 10 points in 100cm^2 at moderate height). This factor is crucial to compensate the influence of non-uniform point distribution in LiDAR data. For fair comparison of the accumulation in different regions, boundary points in sparser regions (lower density) are thus compensated by larger weights, in reverse proportional to their local density. The accumulated effect $C_{hor}(P)$ of horizontal boundary lines to a certain splitting plane P is defined similarly by replacing $\boldsymbol{v}_{ver}$ with $\boldsymbol{v}_{hor} = (0, 1, 0)$.



For ease of discussion, below we first focus on how to place a horizontal splitting plane $P_z := \{\boldsymbol{p} | \langle \boldsymbol{p}, \boldsymbol{v}_{ver} \rangle = z\}$. Note that lower values of $C_{ver}(P_z)$ occur where there is weaker presence of vertical boundary lines, as illustrated in a 2D example on the right, with lower saturation indicating lower value of $C_{ver}(P_z)$. In contrast, $C_{hor}(P_z)$ has higher values at places where horizontal boundary lines are denser. Therefore, to penalize placing of P_z at places with dense vertical lines and/or sparse horizontal lines, the penalty function is formulated as follows:

$$F_z(P_z) = C_{ver}(P_z) - \lambda \cdot C_{hor}(P_z), \quad (2)$$

where λ is a small value ($\lambda \equiv 0.2$ in our experiments), as avoiding vertical boundary lines is more important. We prefer to insert horizontal splitting planes at places with small values of $F_z(P_z)$. Analogously, the penalty function of placing a vertical splitting plane $P_y := \{\boldsymbol{p} | \langle \boldsymbol{p}, \boldsymbol{v}_{hor} \rangle = y\}$ is defined as follows:

$$F_y(P_y) = C_{hor}(P_y) - \lambda \cdot C_{ver}(P_y), \quad (3)$$

with smaller values of $F_y(P_y)$ corresponding to higher preference of placing vertical splitting planes.

Initial Number and Location of Splitting Planes. Let $[x_{min}, x_{max}] \times [y_{min}, y_{max}] \times [z_{min}, z_{max}]$ denote the bounding volume of the current building block. For placing horizontal splitting planes, we first explicitly evaluate the penalty values of $F_z(P_z)$ in the range of $[z_{min}, z_{max}]$ (Figure 4). The corresponding z-coordinates $\{z_1, z_2, ...\}$ of the local minima are then identified as the initial horizontal splitting planes. The strategy of determining the location of vertical splitting planes is identical. As the penalty functions reflect the accumulated effect of vertical and horizontal boundary lines, they can provide generally reasonable initial splitting planes even for highly contaminated data like the one in Figure 4. The number and position of the splitting planes will be refined later in stage 3 (Section 3.4) to further rectify and improve the results.

Direction of Splitting. Penalty functions F_z and F_y can also be used to adaptively determine the optimal splitting direction. Specially, we first evaluate the average of the local minima of F_z and that for F_y. For fair comparison of the two directions with unequal sizes, the average values are thus divided by $\|z_{max} - z_{min}\|$ and $\|y_{max} - y_{min}\|$ respectively. The splitting direction is then chosen as the one with the smaller average penalty.

3.3 Element Grouping

The identified splitting planes subdivide the current building block into slices in the adaptively determined splitting direction. The goal of this step is to group similar slices together, so as to identify repetitive facade elements and split them consistently in the next level. This step also provides the information needed to rectify the splitting planes in the next step. For brevity, our discussion below focuses on the grouping of a series of horizontal slices, since the process of grouping in the vertical direction is identical.

Let $\{z_1, z_2, ..., z_k\}$ denote the identified k horizontal splitting planes organized in an ascending order and $\{S_1, S_2, ..., S_{k+1}\}$ the split $k+1$ horizontal slices, with the z-coordinate of slice S_i falling in the range of $[z_{i-1}, z_i]$ (with $z_0 = z_{min}$ and $z_{k+1} = z_{max}$). The grouping is achieved by a greedy bottom-up clustering algorithm, where the similarity measure between pairs of slices plays an important role.

To measure the similarity between slices S_i and S_j, we first align them together by computing the optimal rigid transformation T_{ij} using a standard ICP algorithm [Besl and McKay 1992]. Here, T_{ij} is restricted to be a translational transformation (i.e., capturing z-component for horizontal slices and y-component for vertical slices), since the underlying facade structure is governed by axis-aligned grids, possibly with interlacing or concatenation.

The similarity between S_i and S_j is defined over their aligned versions. To tolerate poor quality of input data, we quantize the space of the overlapping region of the aligned slices and measure the overlapping ratio between S_i and S_j in this quantized space as their similarity. Specifically, the aligned slices S_i and S_j are embedded into a volumetric grid whose size is determined by the bounding box B of the overlapping region of S_i and S_j. The grid resolution is set to be $0.2m$ in each dimension. For each voxel, we define two boolean functions $\nu_i(k)$ and $\nu_j(k)$, indicating whether a voxel k is covered by S_i and S_j, respectively: $\nu_i(k) = 1$, if there is enough points from S_i in voxel k (5 used in our experiments); $\nu_i(k) = 0$, otherwise. The overlapping ratio of S_j with respect to S_i in such quantized space can then be defined as $R_{S_i} = \sum_k \nu_i(k)\nu_j(k) / \sum_k \nu_i(k)$. The ratio with respect to S_j is defined similarly. The average value of these two ratios is finally used to measure the similarity of two slices, defined as follows:

$$\chi_{ij} = \frac{1}{2}(R_{S_i} + R_{S_j}), \quad (4)$$

Note that such similarity measure lies in the range of [0, 1], with larger values indicating higher similarity. It also supports partial matching of two slices, since the similarity is defined on the overlapping region of their aligned versions. In practice, to eliminate meaningless partial matching (e.g., piece of narrow flat wall with the margins of a window), slices of small size ($< 1m$) are filtered out, and will not be merged with others or participate in the similarity computation and the grouping process thereafter.

Once the similarity is measured for every pair of slices, we then iteratively cluster the pairs of slices that have the maximum similarity using a greedy bottom-up method until no more clusters can be created. The clustering process is stopped until $\chi_{ij} < \tau_c$ (typically 0.65 used in our examples).

3.4 Splitting Rectification

We use a sufficiently large value for σ (i.e., $\sigma = 0.5m$) in Equation 1, successfully suppressing excessive splitting planes (see supplementary). This is mainly because noise or outliers alone are unlikely to form local minima of the penalty functions given such large kernel size, while architectural elements are typically of size larger than $0.5m$ [Müller et al. 2007]. However, due to the local nature of our splitting strategy, local minima might still be corrupted by noisy structures, possibly leading to imperfect position of splitting planes or even fail to detect desired splitting planes (e.g., Figure 5). In addition, our local splitting strategy might also fail to identify structures which are clear only from a more global perspective (e.g., the two splitting planes at the two sides of the facade). To address the problems, we resort to some global information to first recover the missing splitting planes and then perform a global optimization operation to make (approximately) perfect alignment of the splitting planes. Both operations are based on the grouping information obtained in the previous step. Again, we describe the algorithm in context of the horizontal splitting planes.

The key idea of identifying missing splitting planes for a slice is to borrow splitting planes from the slices within the same group (Figure 5). We observed that slices of smaller sizes are typically more likely to be desired facade elements in the current level (e.g.,

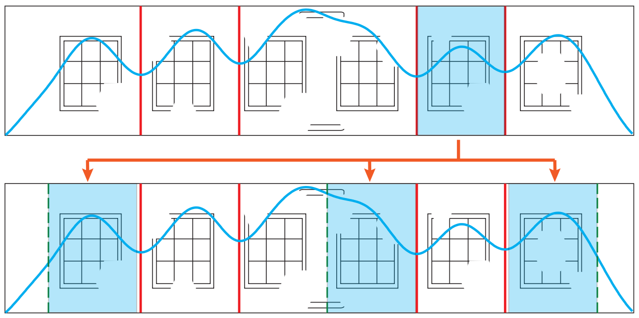


Figure 5: *Adding missing splitting planes (dashed lines) by borrowing splitting planes from the slice of the smallest size.*

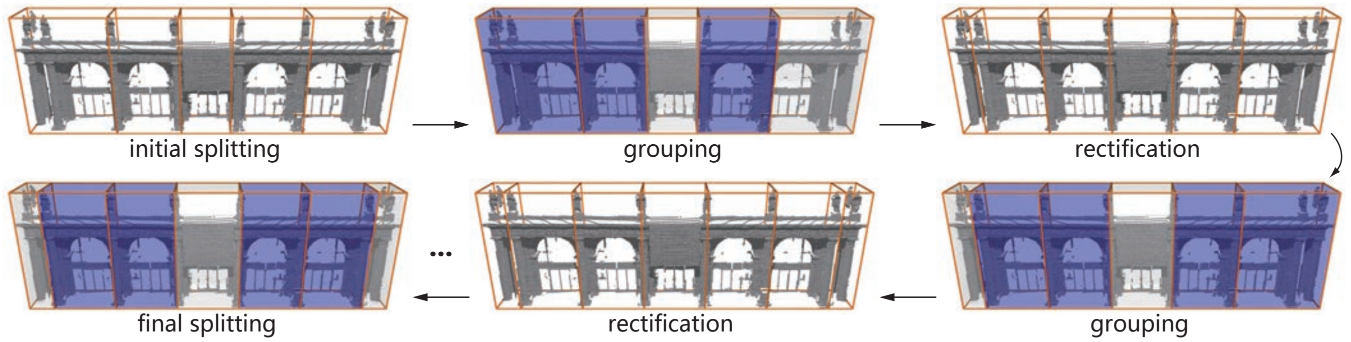


Figure 6: *The iterated grouping and rectification steps for refining the initial splitting planes.*

the highlighted element in Figure 5(top)). For each slice S_i, we thus find the slice S_j that belongs to the same group as S_i and is of the smallest size among all the grouped slices (except for S_i). We then align them together by applying the optimal transformation T_{ji} between S_j and S_i (Section 3.3). Let z^t_{j-1} and z^t_j denote the transformed two boundaries of S_j. An extra splitting plane in S_i is then added in the position of z^t_j (Figure 5(bottom)) if $\|z_i - \min(z_i, z^t_j)\|/\|z_i - z_{i-1}\|$ is above a certain threshold τ_r (typically 25% used in our experiments). Extra slices are created accordingly to replace the original one. The newly created slice corresponding to the overlapping region between S_i and S_j inherits the grouping information of S_i, while the rest of the newly created slices form individual new groups. Similarly, a splitting plane in the position of z^t_{j-1} is added if necessary.

After adding the missing splitting planes, we then intend to refine the position of all the splitting planes using a global optimization. The rationale is to make pairs of already grouped slices aligned as perfectly as possible. The perfect alignment between S_i and S_j means that when S_i and S_j are optimally aligned under T_{ij} (Section 3.3), the corresponding boundaries (i.e., the splitting planes) of the slices should be exactly matched under T_{ij}, i.e., $z_j - z_i = T_{ij}$ and $z_{j-1} - z_{i-1} = T_{ij}$. It naturally leads to the following energy function to be minimized:

$$E_{tar} = \sum_i \sum_j \sum_{k=0}^{1} \delta_{ij}\omega_{ij}\|z'_{j-k} - z'_{i-k} - T_{ij}\|^2, \qquad (5)$$

where $\{z'_0, z'_1, ..., z'_{k+1}\}$ are the new positions of the splitting planes, and $\delta_{ij} = 1$, if S_i and S_j are of the same group; $\delta_{ij} = 0$, otherwise. $\omega_{ij} = \frac{1}{2}(conf(S_i) + conf(S_j))$ is included here to encourage slices with good sampling rate, which is defined as the average of the confidence weight of the two slices. Specifically, similar to the definition of the boundary point confidence (Section 3.1), for each point $\boldsymbol{p}_i$ in a slice, we define $conf(\boldsymbol{p}_i) = \frac{\lambda_1}{\lambda_2}$, which measures the local uniformity of its local neighborhood. The confidence of a slice is then defined as the sum of the point confidence over all the points within the slice.

We introduce another energy so that the splitting planes adhere to their original positions to some extent, which are also weighted by the slice confidence:

$$E_{src} = \sum_i conf(S_i)\|z'_i - z_i\|^2. \qquad (6)$$

The optimization for splitting refinement is finally defined as the following minimization problem:

$$\text{argmin}_{z'_0, z'_1, ..., z'_{k+1}} \alpha E_{tar} + (1-\alpha)E_{src}, \qquad (7)$$

subject to $z'_0 = z_{min}$, $z'_{k+1} = z_{max}$ and $z_i - z_{i-1} = z_j - z_{j-1}$ if S_i and S_j belong to the same group. This makes the two ends of the splitting planes stick to the original positions and slices of the same group have strictly the same size. The weight α ($\equiv 0.9$ in our experiments) balances the influence between the refinement and data terms. The above optimization is a linear least-squares minimization problem and its global solution can be efficiently obtained.

The grouping and rectification steps are interdependent. Thus after the rectification of the splitting planes, the slices are re-grouped again. The grouping and rectification steps are iterated in an alternating manner until convergence. Figure 6 shows an example of the grouping and rectification results after several iterations for a given facade.

3.5 Hierarchy Construction

To construct a hierarchical representation of the entire facade, we start from the whole piece of an input point cloud as the initial building block and apply the adaptive partition operations described in Section 3.2-3.4 *recursively*. Suppose the current building block has been adaptively partitioned to a set of sub-building blocks. To consistently partition each group of repetitive elements in those sub-building blocks, we align the boundaries of adjacent elements and concatenate all the elements in the same groups as a whole for further partitioning. The generated splitting planes are then mapped back to individual elements. The recursive partition process is continued until it reaches a certain prescribed level (2 used in our example), or the width/height of the current building block is below a given threshold ($2m$ used in our examples), or no initial splitting planes can be found in the structure splitting step.

4 Results and Applications

In this section, we first show some partition results on 3D urban facades, together with several applications that can benefit from our approach. We then discuss its extension to image domain for adaptive partitioning of image facades. We have tested our method on various LiDAR scans and facade images of different complexity and styles. In all our examples, we represent the partitioned building blocks using different boxes with repetitive elements sharing the same colors. The running time of our method depends on the number of points (pixels for image extension) in input facades and the levels of recursive partition. For the stop criteria used in this paper, it typically takes around 3 minutes for a facade scan with 300K points and 10 seconds for a facade image in the resolution of 800×600, measured on an Intel Core 2 Duo 3GHz computer with 4GB RAM.

Parameters. Although our method involves a few parameters, only two parameters: τ_c for clustering threshold and τ_r for adding splitting, sometimes need user intervention. The default values of these two parameters typically give good results. However, for very noisy input (e.g., Figure 7(e)), τ_c should be decreased to encourage grouping. Since for certain examples (e.g., Figure 7(a)) there is a

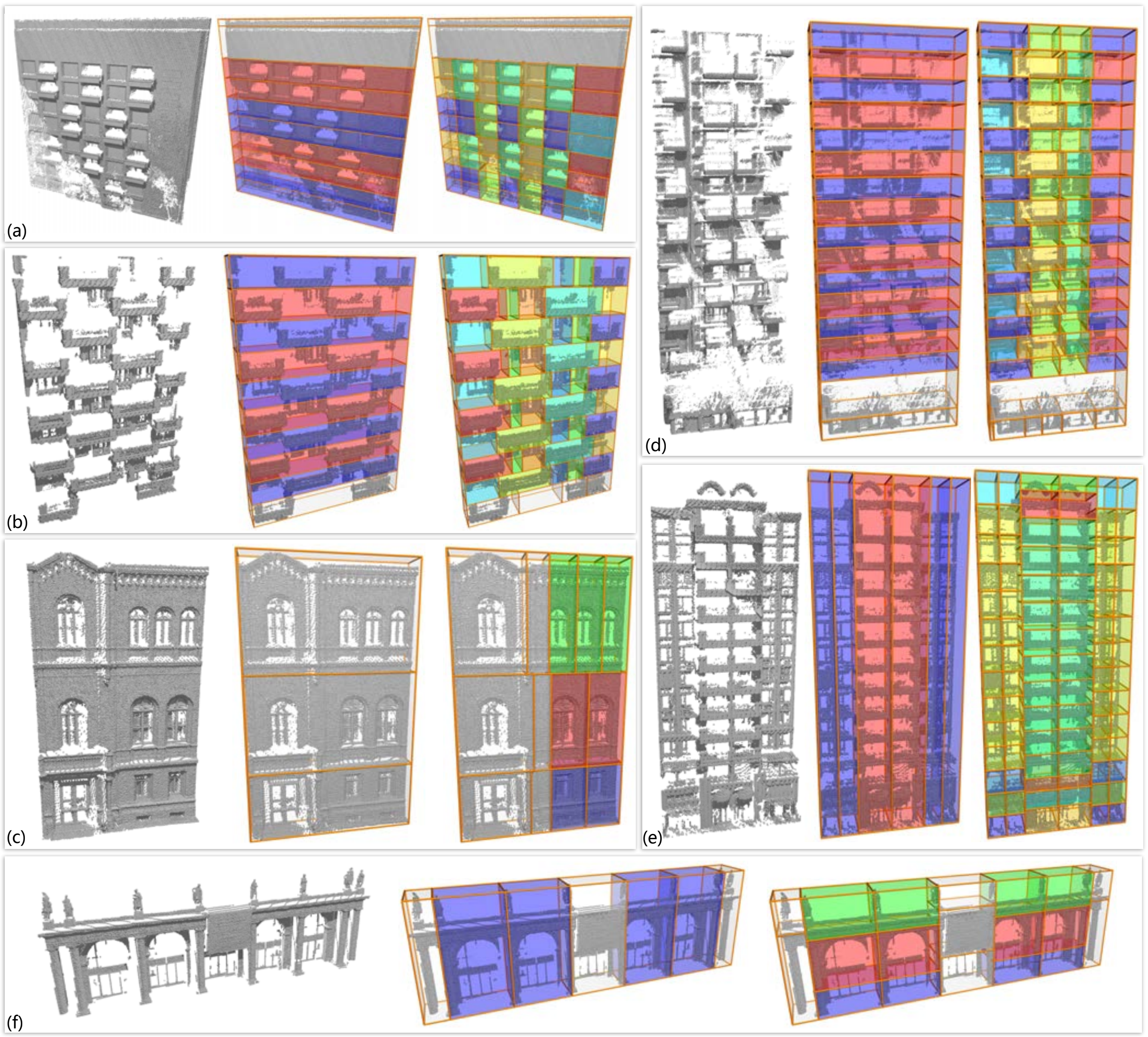


Figure 7: *Result gallery of adaptive partitioning of 3D urban facades with various styles. See additional results at the project page.*

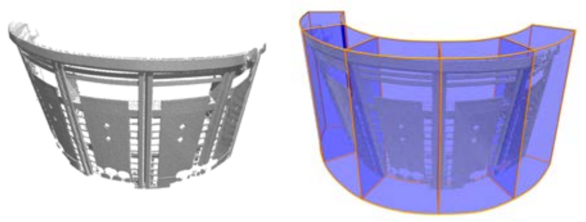


Figure 8: *Extension to an approximately conical facade.*

tradeoff between finding missing splitting and avoiding superfluous splitting, τ_r has to be manually tuned for such cases.

4.1 Adaptive Partitioning of 3D Urban Facades

We first evaluate our method on facades from low-rise buildings. Figure 1(a) gives an example with severe missing data in the lower half. Nevertheless, the hierarchical structure is successfully identified using our top-down approach. A similar example but with quite different grid layout is exhibited in Figure 7(a). Figure 1(b) shows another example, where the facade has multiple groups of repetitive elements. Note that it exhibits several concatenated grids and two sets of grids are separated by another one. Due to the flexibility of our approach, it is capable of discovering such facade structure as well. Our approach is also applicable to facades with only a few repetitions even along in one dimension (e.g., Figure 7(c) and 7(f)), since our method is largely insensitive to the number of repetitions. We then give examples of some high-rise buildings. Figure 1(c) shows the result of applying our method to a tall building. Note that the interlaced floors are discovered in the first level of the hierarchy and the repetitive balconies in the middle part of the facade are found out in the second level. Two other examples are presented in Figure 7(b) and 7(d), which also have interlaced patterns in the vertical direction. Figure 7(e) gives an example with concatenated grids in the horizontal direction. Although we mainly describe our

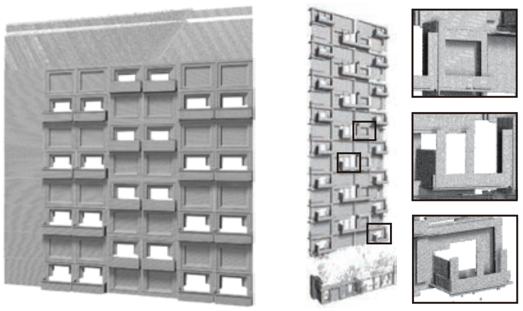


Figure 9: *Automatic non-local consolidation of urban facades.*

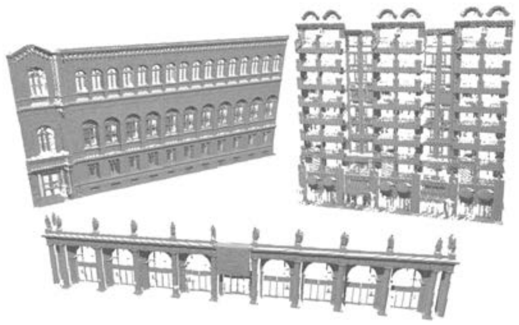


Figure 10: *Manipulation gallery of urban facades.*

algorithm on facades with dominant flat planes, our method can be easily extended to general developable surfaces by changing the parameter domain accordingly, given the dominant planar structure detected. Figure 8 shows an example of extending our method to a facade with approximately conical structure. The individual repetitive tiles along the surface are faithfully partitioned out. Please refer to the project page for additional results.

4.2 Applications

Facade Consolidation. By exploiting our method as an automatic preprocessing tool, the non-local consolidation of urban facades [Zheng et al. 2010] can be performed in a fully automatic manner. Specifically, we use the automatically extracted repetitive building blocks as the input to the main stages of non-local consolidation. Figure 9 shows two examples of applying such a mechanism to the point clouds in Figure 1(a) and Figure 7(d).

Facade Manipulation. After adaptive partitioning, a set of shape grammars can be inferred from the input facades by tracing the partition process and exploiting the repetitive building blocks, as is close in spirit to [Bokeloh et al. 2010]. The original urban facade can then be intelligently manipulated by instancing the shape grammars once again. Figure 10 shows the manipulation gallery of several facade models presented in this paper.

Facade Admixture. Since the adaptive partition process splits the whole piece of facade into a hierarchy of architectural elements, these elements can be combined together to create urban facades with mixed features. Figure 11 shows two such admixture examples: the mixture of elements of Figure 1(a) and Figure 1(b), and the replacement of a group of windows in Figure 1(a) with elements from Figure 1(c). In all these examples, the size of the facade elements is automatically adjusted to fit the target positions.

4.3 Extension to Image Domain

Our method can also be directly extended to image domain for flexible subdivision of facade images. Compared with the above 3D version, adaptive partitioning of facade images can be carried out

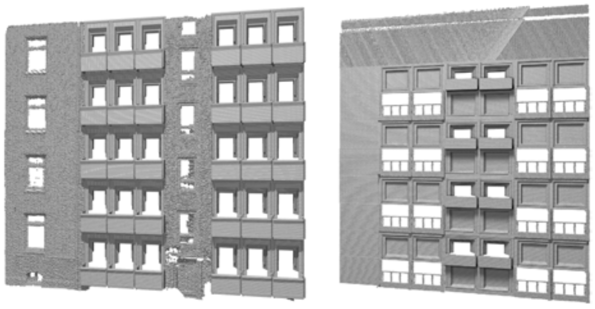


Figure 11: *Admixture of different elements of urban facades.*

with the following modifications: a) In the preprocessing step, we first rectify the input facade image using the approach described in the appendix of [Müller et al. 2007]. The boundary points are then found out using the Canny edge detector [Canny 1986]. The direction and confidence weight of each boundary point are estimated using the same approach described in Section 3.1. b) For computing the penalty functions on images, we simply set $\rho(\boldsymbol{p}_i) = 1$ in Equation 1, due to the uniform sampling of images. c) To register and compare two image slices, we search for the optimal translation that maximizes the normalized cross correlation [Lewis 1995] of their overlapping region. Two image slices are considered repetitive instances and thus grouped together if that maximized normalized cross correlation is above a given threshold (typically 0.7 used in our examples). In this paper, we use the Canny edge detector and normalized cross correlation mainly as a proof of concept to demonstrate the effectiveness of our framework. We expect improvement if more advanced techniques are adopted.

Figure 14 shows some results of adaptive partitioning of facade images. Our approach successfully identifies the underlying rectilinear mixture model in those images. Note that our method is also applicable to facades with round windows, which do not have strong vertical/horizontal lines, since we have already taken into account boundary points of different directions in the computation of the accumulation (via inner product in Equation 1).

To further evaluate the effectiveness of our approach, we tested our method on the *eTRIMS Image Database* [Korč and Förstner 2009], which is composed of 60 facade images of various styles and configurations. We provide the results of all the 60 examples at the project page, including intermediate results like initial splitting. For 48 out of the 60 tested images, our automatic processing method produces satisfactory results. For the rest of the images, the results are more or less problematic, mainly caused by strong mirror reflection, dark shadow, extensive sundries (e.g., flowerpot) around the windows (which could confuse the grouping of image patches in our current implementation), and strong self occlusion due to significant deviation of the shooting angle (which could lead to interleaved edges and thus influence the splitting step).

Our method has also been tested on a more challenging data set, *CVPR 2010 data set* in the *Ecole Centrale Paris Database* [Teboul et al. 2010][1]. See a few representative results at the project page. Although both the eTRIMS and CVPR 2010 data sets contain images of street scenes captured in European cities, the facades in the latter data set are of more classic architectures and many of them have Haussmannian or similar building style. Such classic architectures often contain excessive ornaments between tiles in certain floors (Figure 13), e.g., balcony railings spanning entire floors or repeated decorative lines between facade elements. Those ornaments might introduce undesired edges and thus confuse our technique. Although our method is generally able to successfully di-

[1]Data source: http://www.mas.ecp.fr/vision/Personnel/teboul/data.php

vide facades into floors, it is less robust for partitioning floors into horizontal tiles due to such undesired edges (see Figure 13 middle and right). Around 20% of the results are problematic. On the other hand, unlike the facades in the eTRIMS database, which vary more in architectural structure, the Haussmannian or similar building facades often carry a more consistent typology and composition rhythm, and present rather regular elements. Therefore the techniques that explicitly incorporate the architectural knowledge of the Haussmannian style [Liu and Gagalowicz 2010] or use prescribed shape grammars [Teboul et al. 2011; Teboul et al. 2010] might be more effective and stable.

Comparison with Previous Work. In terms of algorithm design, our work is the most related to that of [Müller et al. 2007]. Since our method is designed based on a weaker assumption of facade structure, it can generate more flexible facade subdivision. Figure 14 shows the results of such comparison. Note that some tiles are over-partitioned by [Müller et al. 2007] due to their global rectilinear grid assumption. In contrast, our approach is capable of subdividing them into reasonable hierarchal structures due to its adaptability.

4.4 Limitations

Our method is not applicable to facades which cannot be described by our rectilinear mixture model, e.g., facades with completely irregular repetitive patterns (Figure 12(left)). In our current implementation, tiles that are essentially similar may not be grouped if their parent slices have already been separated into different groups in the previous level, such as the facade with round windows in the middle of Figure 14. Such problem might be addressed by an extra bottom-up grouping (post-)process. Our approach is purely geometric, possibly leading to imperfect partitioned building blocks in some cases (e.g., the right-most balconies in Figure 1(c)). This is mainly due to the existence of the extra dense vertical boundaries along the current splitting plane. Although our method is applicable to facades with few or no repetitive elements (e.g., Figure 12(right)) given our focus on splitting instead of searching for repetitive patterns directly (cf. [Wu et al. 2010]), our method is less robust in such cases. Lastly, our system relies heavily on the first splitting step and there is currently no way for recovering from severe errors in this stage (e.g., over-splitting planes). Revisiting this step by employing the grouping information from the second and third steps (potentially demanding an outer iteration) might be a possible direction to address this problem.

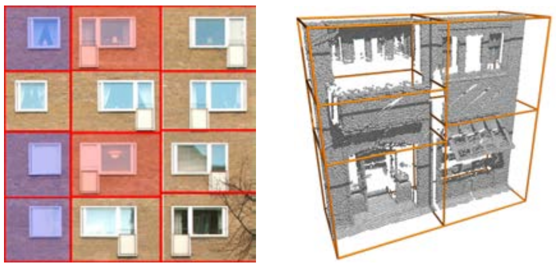


Figure 12: *Two challenging examples.*

5 Conclusion

This paper investigates the problem of generating a flexible and hierarchical representation of urban facades. Our key observation is that facades with concatenated and/or interlaced grids are ubiquitous in urban scenes. We provide an effective solution to this problem by introducing the concept of adaptive partitioning. The splitting direction, number and location of splitting planes are adaptively determined in the structure splitting step. It is then followed by iterated grouping and rectification steps to find repetitive elements and improve the splitting positions. These steps are carried out recursively to obtain a high-level facade structure. Due to the flexibility of our approach, we are able to handle urban facades with various complexity and styles, represented as either 3D facade scans or image facades. We also show a series of applications that can benefit from our method. Currently, our method is purely geometric based. It might be possible to introduce extra prior knowledge obtained from a training process. With the increasing popularity of LiDAR data, that could be an interesting problem to be investigated in the future.

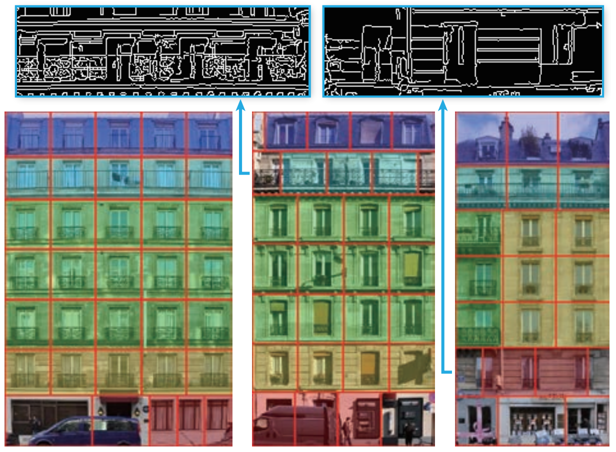


Figure 13: *Testing our approach on the CVPR 2010 data set.*

Acknowledgements

We thank the anonymous reviewers for their constructive comments. The LiDAR scans in Figure 1(b), Figure 7(c)(f) and Figure 12 are courtesy of the Institute of Cartography and Geoinformatics, Leibniz University at Hannover. The rest of the LiDAR scans in the paper are courtesy of Professor Baoquan Chen at Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. Shi-Min Hu is supported by the National Basic Research Project of China (Project Number 2011CB302203), the Natural Science Foundation of China (Project Number 61120106007). Hongbo Fu is partly supported by grants from CityU (Project Number SRG7002533), and the HKSAR Research Grants Council (Project Number 9041562).

References

BECKER, S., AND HAALA, N. 2009. Grammar supported facade reconstruction from mobile LIDAR mapping. In *Proc. CMRT. Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci*, 229–234.

BESL, P. J., AND MCKAY, N. D. 1992. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell. 14*, 239–256.

BOKELOH, M., BERNER, A., WAND, M., SEIDEL, H., AND SCHILLING, A. 2009. Symmetry detection using feature lines. *Computer Graphics Forum 28*, 2, 697–706.

BOKELOH, M., WAND, M., AND SEIDEL, H.-P. 2010. A connection between partial symmetry and inverse procedural modeling. *ACM Trans. Graph. 29*, 4, 104:1–104:10.

BUCKSCH, A., LINDENBERGH, R., AND MENENTI, M. 2010. Skeltre: Robust skeleton extraction from imperfect point clouds. *The Visual Computer 26*, 1283–1300.

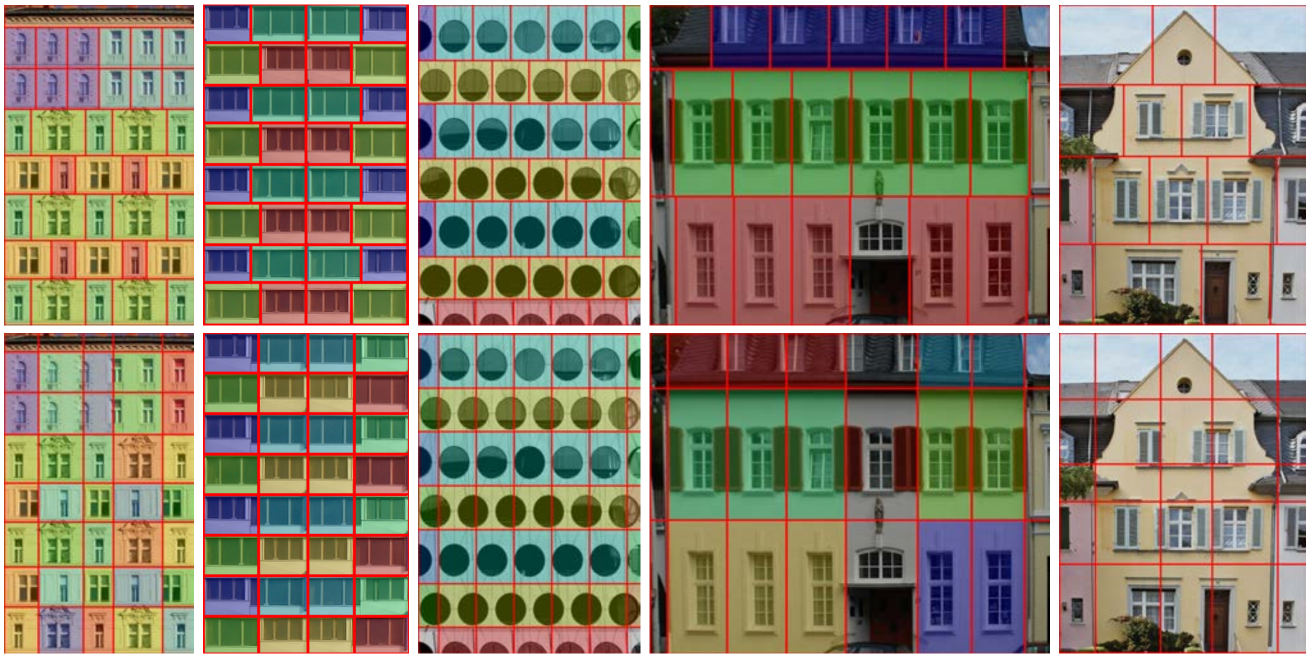


Figure 14: *Adaptive partitioning of facade images and comparison with previous work.* ***First row:*** *ours.* ***Second row:*** *Müller et al.'s. See our results on the eTRIMS image database with 60 facade images at the project page.*

CANNY, J. 1986. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell. 8*, 679–698.

KORČ, F., AND FÖRSTNER, W. 2009. eTRIMS Image Database for Interpreting Images of Man-Made Scenes. Tech. Rep. TR-IGG-P-2009-01, March.

LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J., AND FULK, D. 2000. The digital Michelangelo project: 3D scanning of large statues. In *Proceedings of SIGGRAPH '00*, 131–144.

LEWIS, J. 1995. Fast normalized cross-correlation. In *Vision Interface 1995*, 120–123.

LIU, C., AND GAGALOWICZ, A. 2010. Image-based modeling of haussmannian facades. *International Journal of Virtual Reality 9*, 1, 13–18.

MITRA, N. J., GUIBAS, L. J., AND PAULY, M. 2006. Partial and approximate symmetry detection for 3d geometry. *ACM Trans. Graph. 25*, 3, 560–568.

MÜLLER, P., WONKA, P., HAEGLER, S., ULMER, A., AND VAN GOOL, L. 2006. Procedural modeling of buildings. *ACM Trans. Graph. 25*, 3, 614–623.

MÜLLER, P., ZENG, G., WONKA, P., AND VAN GOOL, L. 2007. Image-based procedural modeling of facades. *ACM Trans. Graph. 26*, 3, 85:1–85:10.

MUSIALSKI, P., RECHEIS, M., MAIERHOFER, S., WONKA, P., AND PURGATHOFER, W. 2010. Tiling of ortho-rectified facade images. In *Spring Conference on Computer Graphics*.

NAN, L., SHARF, A., ZHANG, H., COHEN-OR, D., AND CHEN, B. 2010. Smartboxes for interactive urban reconstruction. *ACM Trans. Graph. 29*, 4, 93:1–93:10.

NING, X., ZHANG, X., AND WANG, Y. 2010. Automatic architecture model generation based on object hierarchy. In *ACM SIGGRAPH Asia 2010 Research sketches*.

PAULY, M., MITRA, N. J., WALLNER, J., POTTMANN, H., AND GUIBAS, L. J. 2008. Discovering structural regularity in 3d geometry. *ACM Trans. Graph. 27*, 3, 43:1–43:11.

SCHNABEL, R., WAHL, R., AND KLEIN, R. 2007. Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum 26*, 2, 214–226.

TEBOUL, O., SIMON, L., KOUTSOURAKIS, P., AND PARAGIOS, N. 2010. Segmentation of building facades using procedural shape priors. In *CVPR 2010*, 3105–3112.

TEBOUL, O., KOKKINOS, I., SIMON, L., KOUTSOURAKIS, P., AND PARAGIOS, N. 2011. Shape grammar parsing via reinforcement learning. In *CVPR 2011*.

WONKA, P., WIMMER, M., SILLION, F., AND RIBARSKY, W. 2003. Instant architecture. *ACM Trans. Graph. 22*, 3, 669–677.

WU, C., FRAHM, J., AND POLLEFEYS, M. 2010. Detecting large repetitive structures with salient boundaries. *ECCV 2010*, 142–155.

XIAO, J., FANG, T., TAN, P., ZHAO, P., OFEK, E., AND QUAN, L. 2008. Image-based façade modeling. *ACM Trans. Graph. 27*, 5, 161:1–161:10.

XIAO, J., FANG, T., ZHAO, P., LHUILLIER, M., AND QUAN, L. 2009. Image-based street-side city modeling. *ACM Trans. Graph. 28*, 5, 114:1–114:12.

YAMAZAKI, I., NATARAJAN, V., BAI, Z., AND HAMANN, B. 2010. Segmenting point-sampled surfaces. *The Visual Computer 26*, 12, 1421–1433.

ZHENG, Q., SHARF, A., WAN, G., LI, Y., MITRA, N. J., COHEN-OR, D., AND CHEN, B. 2010. Non-local scan consolidation for 3d urban scenes. *ACM Trans. Graph. 29*, 4, 94:1–94:9.