DOI: 10.1111/j.1467-8659.2012.03046.x
EUROGRAPHICS 2012 / P. Cignoni, T. Ertl
(Guest Editors)

*Volume 31 (2012), Number 2*

# Factored Facade Acquisition using Symmetric Line Arrangements

Duygu Ceylan — EPFL
Niloy J. Mitra — UCL
Hao Li — Columbia University
Thibaut Weise — EPFL
Mark Pauly — EPFL

**Abstract**

*We introduce a novel framework for image-based 3D reconstruction of urban buildings based on symmetry priors. Starting from image-level edges, we generate a sparse and approximate set of consistent 3D lines. These lines are then used to simultaneously detect symmetric line arrangements while refining the estimated 3D model. Operating both on 2D image data and intermediate 3D feature representations, we perform iterative feature consolidation and effective outlier pruning, thus eliminating reconstruction artifacts arising from ambiguous or wrong stereo matches. We exploit non-local coherence of symmetric elements to generate precise model reconstructions, even in the presence of a significant amount of outlier image-edges arising from reflections, shadows, outlier objects, etc. We evaluate our algorithm on several challenging test scenarios, both synthetic and real. Beyond reconstruction, the extracted symmetry patterns are useful towards interactive and intuitive model manipulations.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling— I.4.5 [Image Processing and Computer Vision]: Reconstruction—

## 1. Introduction

Reconstruction of geometric models from a small set of images is an easy, flexible, and economic method for large scale 3D content creation. The simplicity of the acquisition process, however, comes with stronger demands on the processing algorithms. Fundamentally, any such algorithm that uses triangulation to infer 3D information from the images has to address the difficult and often ambiguous correspondence problem, i.e., identify the point-pairs that represent the same world space location between any image pair.

Advances in camera technology and multi-view stereo methods have lead to significant improvements in the quality of the reconstructed models (see [FP09, AFS*10] and references therein). Despite this success, many challenges remain in the acquisition and reconstruction of clean, precise, and high-quality models of complex 3D objects, in particular when intuitive post-processing and editing of the acquired geometry is desired (see survey [VAW*10]).

Most multi-view stereo (MVS) methods use local feature or window-based matching in combination with local smoothness priors to produce 3D samples. Such local processing makes it fundamentally difficult to resolve ambiguities and can lead to high noise levels and a significant amount of outliers. This is particularly true for models with a large number of repetitive elements, where stereo methods are easily confused due to a multitude of locally consistent

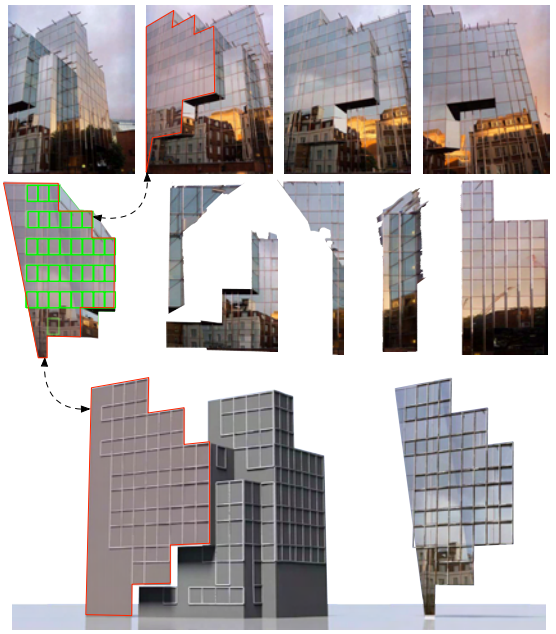

**Figure 1:** *The integration of structure-discovery and line-based 3D reconstruction yields high-quality 3D models even for complex surface materials that pose severe problems for existing methods. (Bottom-right: shows geometry+texture for one plane to indicate alignment quality.)*

feature matches. On the other hand, for models with strong reflective faces, local-correlation based matching often produces only a sparse set of points as output (see Figure 13).

We address these shortcomings and introduce a novel approach for image-based reconstruction of piecewise planar objects containing symmetric parts, such as building facades (see Figure 1). Specifically, we incorporate structural priors at two levels: (i) We fit geometric primitives, such as lines and planes, to capture small and medium scale spatial coherence in the data, and (ii) we extract reflective, translational, and rotational repetitions to provide reconstruction priors that exploit non-local coherence. These priors explicitly capture the dominant symmetries of the acquired object. While line and plane features represent continuous symmetries, repetitive elements model discrete symmetries.

We simultaneously operate on the input 2D images and intermediate 3D reconstructions, and couple the two using the extracted symmetries. Note that the symmetry priors are not specified a priori, but are directly learned from the input. Finding stable symmetries in 3D data, however, is a difficult problem, particularly from sparse and incomplete MVS data. In practice we face a cyclic dependency: to remove noise and outliers and fill holes, we need to find reliable symmetries; yet to robustly estimate symmetries, we need clean and complete data sets. We bootstrap the process by allowing the user to roughly indicate an arbitrary repeated instance on one of the images. Subsequently, we formulate a combined reconstruction-detection algorithm that iteratively propagates geometric and structural information to reinforce symmetries and 3D sample locations.

Intuitively, we exploit large scale symmetries among linear elements, e.g., window frames on building facades, to improve reconstruction quality. In traditional MVS reconstruction, widely spread repetitions can be a source of confusion: in wide-baseline MVS, such repetitions can result in large misalignments due to ambiguous matches; alternately, employing a series of narrow-baseline symmetries leads to accumulation error during the stitching phase. Instead, we exploit symmetries as non-local priors via a coupled symmetry extraction and 3D reconstruction to produce high quality outputs starting from noisy and sparse linear features. Note that, unlike other approaches such as [FCSS09a], we do not make a Manhattan assumption or expect the model to be axis-aligned (see Figure 2). The final output explicitly encodes the detected repeated structures producing a *factored facade* model, making subsequent image- or model-space editing operations easy and intuitive.

We evaluate our framework on a range of synthetic and real scenarios, under large scale reflections, spurious objects, and strong shadows. Since the success of our approach depends on symmetries in the acquired object, we focus on man-made structures such as architectural scenes that are important for large-scale urban reconstruction. Our method is not targeted towards organic shapes such as trees or other highly irregular objects. For such objects, our framework effectively degenerates to traditional MVS reconstruction.

**Contributions:** Our key contribution is the integration of structure discovery and geometry consolidation into a 3D reconstruction algorithm for urban scenes from images. The main technical novelty is a coupled optimization that combines low-level geometric feature extraction with symmetry detection both on the input 2D images and intermediate 3D representations. Detected symmetries are used to iteratively refine the confidence and spatial location of initially unreliable geometric features, which in turn leads to more complete and precise symmetry transformations. As a result, we obtain consistent and accurate 3D models that explicitly represent the semantic structures of the acquired buildings, which is beneficial for rendering and post-editing.

## 2. Related Work

Fast and accurate reconstruction of urban facades and buildings has received significant attention from researchers in computer vision and computer graphics. We discuss the main strategies that have been explored.

**Multi-view stereo (MVS):** A multitude of successful MVS algorithms have been developed in recent years [SCD*06, GSC*07, FP09]. These approaches naturally benefit from the continuous improvement and increase of resolution of digital cameras. With modern GPUs some of these algorithms even run in real-time. For the reconstruction of man-made objects an additional prior may be incorporated: most objects in the scene consist of piecewise planar elements. Mičušík et al. [MK10] employ a super-pixel segmentation approach with MRF-plane labeling instead of pixel-wise depth labeling for more accurate and consistent reconstructions. Werner et al. [WZ02] additionally try to fit specific models such as roof windows or doors. Based on the even stronger assumption that all planes are axis-aligned (Manhattan-world), Furukawa et al. [FCSS09a] propose a method for image-based modeling ensuring local photometric consistency and enforcing global visibility constraints. Subsequently, they extend the method to model building interiors [FCSS09b]. Potential plane candidates are either found using vanishing lines [KZ02] or estimated from a sparse point-cloud reconstruction [FCSS09a]. This typically works well for the three dominant directions, but small local planar regions are often lost. We therefore use 3D lines [SZ97, WZ02] that can be reconstructed reliably and also allow the recovery of smaller planar patches without imposing an orthogonality constraint that would often be violated in real scenes (see Figure 2).

**Procedural modeling:** In one of the early efforts, Wonka et al. [WWSR03] use split grammars and an attribute matching system to synthesize buildings with a large variety of different styles. Given architectural footprints, Kelly et al. [KW11] demonstrate how buildings can be interactively modeled using procedural extrusions. These methods, how-

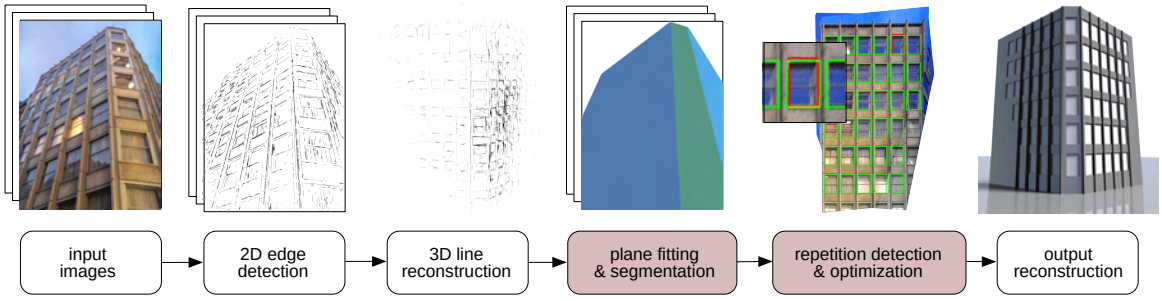


**Figure 2:** *Algorithm pipeline. We first perform 2D edge detection on a set of input images. Edge-based stereo matching then yields a sparse set of consistent 3D lines, from which a set of candidate planes for image segmentation is computed. For each plane, a joint optimization refines the parameters for the candidate line segments and their coupling symmetry transforms to produce a symmetry-factored 3D reconstruction. Stages in red optionally involve user guidance.*

ever, are not directly useful for model acquisition. Earlier, Müller et al. [MZWG07] explore auto-correlation based analysis of rectified images combined with shape grammars towards urban reconstruction. They propose a clever mix of user interaction and image analysis for rule-based procedural modeling. The method, however, fails to handle large differences due to reflective surfaces or interleaved repetitions (e.g., multiple repetitive patterns in Figure 10).

**3D guided model synthesis:** Multiple data sources (e.g., photographs, LiDAR scans, aerial images, GIS data) have been combined to improve the quality of 3D models [FJZ05, LZS*11]. Such methods, however, require precise alignment across the multiple modes making data acquisition challenging. Directly working with incomplete LiDAR scans, Zheng et al. [ZSW*10] use model scale repetitions to create consolidated point clouds, while Nan et al. [NSZ*10] propose an interactive framework for quick architectural modeling using 3D point cloud data for guidance. One can directly detect symmetries from good quality 3D inputs using a transform domain analysis [PMW*08], slippable features [BBW*09], or learned line features [SJW*11], but the methods fail on sparse MVS point sets. Instead, we couple MVS reconstruction and symmetry recovery through a tight 2D-3D optimization to produce high quality outputs, even when using only a handful of input images.

**Interactive model synthesis:** Debevec et al. [DTM96] use manually marked lines in photographs for image-based modeling of buildings. Chen et al. [CKX*08] interpret freehand sketches to create texture-mapped 2.5D building models using a database of geometric models to fill in plausible details. Sinha et al. [SSS*08] present an interactive system to generate textured piecewise-planar 3D models of urban buildings from unordered photo-collections based on information from structure from motion on the input photographs. Similarly, Xiao et al. [XFT*08] use the output of a MVS system for facade reconstruction in street-level imagery. In their follow-up work [XFZ*09] they replace the necessary interactive strokes by a fully automatic scheme. Image level translational symmetry has also been used for non-local image repair [MWR*09]. Wu et al. [WFP11] demonstrate that repetitive structures can be used for dense reconstruction from a single image by directly enforcing depth consistency between repetitive structures during the optimization (see Figure 12). Similarly, Jiang et al. [JTC09] perform camera calibration from a single image by exploiting symmetry, and allow the user to interactively annotate architectural components using the reconstructed 3D points as anchors, producing a textured polygonal reconstruction. In contrast, we require a few guiding strokes from the users to resolve ambiguity, while the coupled 2D-3D optimization enables a factored facade level model reconstruction.

## 3. Overview

We first provide an overview of our processing pipeline as illustrated in Figure 2. Our algorithm takes as input a set of images of a (static) 3D scene. We start by performing image-space edge detection on each individual image. These detected 2D edges, however, do not all correspond to relevant 3D edges of the geometry. They contain many outliers arising due to shadows, texture patterns, occluding objects, reflections, or depth discontinuities (see Figure 3). We therefore collect the 2D line features across all the images and apply multi-view stereo matching directly on the lines to obtain a set of candidate 3D line features. Note that in this stage we aggressively prune out potential mismatched edges, and later recover and consolidate the edges in the symmetry-based inference stage. Yet we retain sufficiently many correct 3D lines to create a set of candidate 3D planes that allow a consistent segmentation of the images. Using the planes as labels, we employ a Markov Random Field (MRF) formulation that directly incorporates the feature lines by assigning attribute data costs only to the pixels corresponding to the projections of the 3D lines. Symmetry detection is then performed on the projections of these 3D lines onto the planes, using a line- and region-based correlation with respect to in-plane translations and reflections.

At this stage, we simply extract a collection of candidate

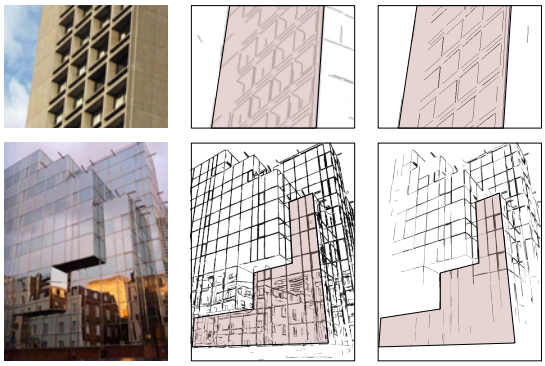

**Figure 3:** *Edges detected directly on images often contain outliers from reflections, shadows, occluding elements, etc. (middle). We employ edge-level MVS to prune out such outliers (right). Note that we also lose many valid edges, which are later recovered via a symmetry-aware optimization.*

line segments approximately coupled by the corresponding symmetry transforms. Subsequently, in a crucial step, we simultaneously refine the parameters for the candidate line segments and their coupling symmetry transforms via an iterative optimization. The resultant factored symmetric parts still lie on the corresponding embedding planes and lack depth variations. Hence, based on user prescribed procedural depth offsets, we extrude or retract the corresponding features to fine tune their depth offsets. Finally, we explore a range of editing options, where symmetric parts are non-locally coupled, using the recovered factored symmetry representation of the input models.

## 4. Algorithm Details

In this section we provide details for the individual stages of our pipeline. From the given set of input images $\mathcal{I} := \{I_1, \dots I_n\}$ we first estimate camera calibration information using the method of Snavely et al. [SSS06]. We perform image-space edge detection (we use open-source EdgeLink[†] based on Canny edge detector) on each individual image $I_i$ to get a collection of 2D edges $\mathcal{L}^2(I_i)$. Multi-view stereo matching using edge-based consistency validation then produces a set of candidate 3D line segments $\mathcal{L}^3 := \{l_1, l_2, \dots\}$.

**4.1 Candidate Plane Construction:** By searching for sets of coplanar lines in $\mathcal{L}^3$, we compute a set $\mathcal{P}$ of candidate planes for 3D model segmentation. We test if line segments $l_i := (\mathbf{v}_1, \mathbf{v}_2)$ and $l_j := (\mathbf{v}_3, \mathbf{v}_4)$ are coplanar as

$$\left(\frac{\mathbf{v}_1+\mathbf{v}_3}{2} - \frac{\mathbf{v}_2+\mathbf{v}_4}{2}\right)^T \frac{(\mathbf{v}_1-\mathbf{v}_3)\times(\mathbf{v}_2-\mathbf{v}_4)}{\|(\mathbf{v}_1-\mathbf{v}_3)\times(\mathbf{v}_2-\mathbf{v}_4)\|} \approx 0, \quad (1)$$

i.e., the diagonals of quad $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4)$ intersect (we test for both orders of $\mathbf{v}_3, \mathbf{v}_4$ and $\mathbf{v}_4, \mathbf{v}_3$). Initially we mark all lines as unclaimed. If a randomly selected unclaimed line pair passes the coplanarity test, we compute the corresponding plane normal as $\mathbf{n} = (\mathbf{v}_1 - \mathbf{v}_2) \times (\mathbf{v}_3 - \mathbf{v}_4) / \|(\mathbf{v}_1 - \mathbf{v}_2) \times (\mathbf{v}_3 - \mathbf{v}_4)\|$ and intercept as $d = -\mathbf{n}^T \mathbf{v}_1$. If such a plane $P_{ij}$ has *sufficient* inlier (witness) lines $l \in \mathcal{L}^3$ that satisfy $l \in P_{ij}$, we include $P_{ij}$ in the set $\mathcal{P}$ of candidate planes and mark the detected inlier lines as *claimed*. In our experiments, we used an inlier count greater than $3-5\%$ to mark sufficiency. We unmark any claimed lines if they are close to the intersection of plane pairs in $\mathcal{P}$ since such lines can potentially belong to multiple planes.

**4.2 Consistent Image Segmentation:** We now segment each image $I_i$ using the 3D line set $\mathcal{L}^3$ and plane set $\mathcal{P}$ by assigning the pixels of $I_i$ to the most likely planes in $\mathcal{P}$. For this purpose, we introduce two terms for each pixel $p \in I_i$:

*(i) Data term:* We note that edge-based multi-view stereo matching helps to distinguish between pixels for which robust depth estimates can be made. Specifically, pixels that do not lie on the projected lines from $\mathcal{L}^3$ are more likely to be found in regions where stereo matching fails. Hence, if $p \notin l \ \ \forall l \in \mathcal{L}_i^{3\to 2}$, we set $E_{data}(p, \mathcal{P}_j) = e^{0.0} \ \ \forall \mathcal{P}_j \in \mathcal{P}$. On the other hand, it is easy to estimate the inconsistency of plane labelings for the pixels that lie on the projected lines. Therefore, if $p \in l$ for some $l \in \mathcal{L}_i^{3\to 2}$ we can deduce the 3D position of this pixel from the line $l$. Now for each plane $\mathcal{P}_j \in \mathcal{P}$ we project this 3D point onto the plane $\mathcal{P}_j$ say $\mathbf{p}'_j$, and then project $\mathbf{p}'_j$ back to the image $I_i$ to get a 2D coordinate $p''_j$. If $d(p, p'') > r$, where $d(p, p'')$ denotes 2D Euclidean distance and $r$ is a threshold distance (usually set to 1% of image width in our experiments) the plane assignment is inconsistent with the known 3D position: so we set a high data cost, $E_{data}(p, \mathcal{P}_j) = e^{1.0}$. If $d(p, p'')$ is below the threshold, we use a multi-view photo-consistency measure similar to Sinha et al [SSS09]. We project the 3D point $\mathbf{p}'_j$ to neighboring views of $I_i$ and set the data cost $E_{data}(p, \mathcal{P}_j) = e^{-s}$ where $s$ is the (average) NCC matching score of local windows centered at $p$ in image $I_i$ and the projection pixels in the neighboring views.

*ii) Smoothness term:* We use a smoothness term to enforce consistent plane neighbors [FCSS09a]. For this purpose, not all lines in any set of coplanar lines in $\mathcal{L}^3$ are relevant — we only expect abrupt changes in pixel labeling across intersections of planes. Hence, we leave out lines from $\mathcal{L}^3$ that do not lie near intersections of plane pairs from $\mathcal{P}$ (with slight abuse of notation we still call this reduced set $\mathcal{L}^3$). For any neighboring pixels $p, q$ we add a constant-weight smoothness term, but set the smoothness weight to zero if the points $p, q$ lie on two sides of a line in $l \in \mathcal{L}^{3\to 2}$ (to handle rasterization, we work with a small approximation margin). We combine the two terms and solve the pixel-plane labeling problem using a standard Markov Random Field (MRF) formulation based on the sequential tree-reweighted message passing algorithm [Kol06].

Once the plane labeling has been computed for each image, we project the 3D lines to the segmented images and

[†] http://www.csse.uwa.edu.au/∼pk/research/matlabfns/#edgelink.

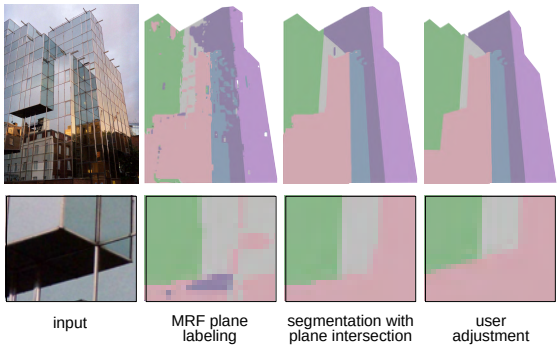


**Figure 4:** *We use an MRF-based plane labeling method to segment the input images. The segmentation is cleaned using the intersection lines obtained by neighboring 3D planes. User input resolves regions of insufficient feature lines.*

extract accurate edges across noisy segment boundaries (see Figure 4). At this stage, if necessary the user can manually make corrections on wrong segment boundaries that may arise due to insufficient reliable 3D lines. These corrections are made by sketching rough strokes on the images which are then snapped to the 2D edges to form the final image segmentation. This new boundary is propagated across the images using the 3D plane and the calibration. Finally, for each candidate plane the corresponding image segments are composited to produce a rectified plane texture. During this composition, for each plane, images that have a smaller foreshortening factor are favored to discard any possible inconsistency in the assignment of the plane across the images.

**4.3 Symmetry-based Optimization:** For each 3D candidate plane $\mathcal{P}_j \in \mathcal{P}$ a rectified texture is composited as described in the previous section. We perform in-plane symmetry detection and refinement for each of the rectified composites. In cases of simple facades, we can automatically detect symmetric elements using current methods [WFP10, JTC11]. However, often there is ambiguity in the choice of repeated elements and the relevant scale (see Figure 6). Further, image-based similarity measurements become unreliable for reflective or textureless surfaces. Hence, we use a few user-annotated rough strokes denoting elements of interest to search for similar elements across the (rectified) images (Figure 5). We perform the image-level matching using a combination of two attributes: (i) normalized cross-correlation (NCC) to compare the local images based on the user-marked region as a template, and (ii) the extracted edges to compare the gradient maps. In this initialization step, we use 2D edges obtained by projecting the 3D lines onto the rectified image. Let $\{l_i^t\}$ be the lines from the user (template) strokes and let $\mathcal{L}' := \{l_j'\}$ denote all the lines containing the edge segments of the projected 3D lines. For each line $l^t \in \{l_i^t\}$ we select all lines $l \in \mathcal{L}'$ with $dist(l, l^t) \leq \varepsilon$ using a suitable threshold $\varepsilon$. The distance is measured in the line-space parameterized by $(\mathbf{n}, d)$ (see [LZS*11]). The selected lines are then projected onto $\{l_i^t\}$. We measure the percentage of the line lengths covered by the feature lines in $\mathcal{L}'$ and compute a similarity score as a weighted combination of line compatibility and (absolute) NCC scores. We accept a match if this final score is above a threshold (0.8 in our experiments; In the table and mirrored glass building examples in Figure 10 we only used line matching scores).

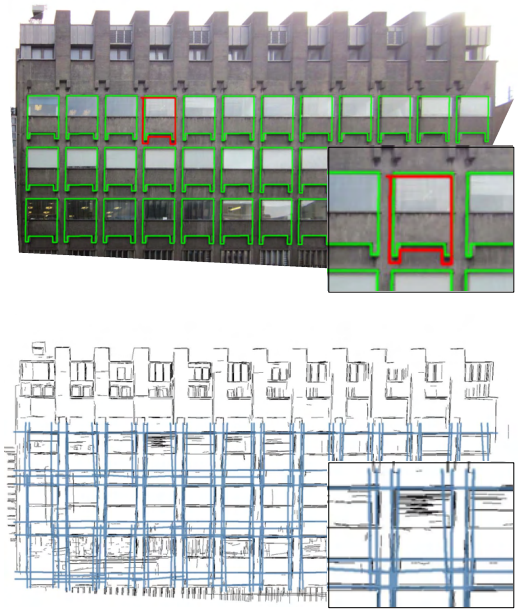

**Figure 5:** *A rough sketch by the user (red) is computationally refined using a symmetry guided optimization allowing robust extraction of useful features (green) even in presence of significant outliers and approximate features.*

In man-made objects, especially in building facades, regularity is predominant not only across element-pairs but also across their mutual arrangement – typically in the form of 1D and 2D grid-like arrangements. This is not surprising since architectural guidelines give strong preference to such grid-structures both for aesthetic and economic considerations [DS08]. We use a non-linear grid fitting approach to generate estimates of corresponding grid generators along with the potential repeated elements [PMW*08]. Note that at this stage certain grid elements can be missing. These are recovered in a later stage.

**Symmetry refinement:** For any rectified plane, let $\{l_1, l_2, \ldots\}$ denote a collection of lines linked by an initial symmetry estimate $T$, i.e., $l_i \approx T^{i-1}(l_1)$ and let line candidate $l_i$ be represented in the normal-intercept form as $l_i := \{\mathbf{p} | \mathbf{n}_i^t \mathbf{p} + d_i = 0\}$. At this stage both the line parameters $\{(\mathbf{n}_i, d_i)\}$ and the estimated transform generator $T$ are imprecise, and our goal is to improve the initial estimates of the lines and the coupling transform. Subsequently, we use the consolidated information to perform a symmetry-guided search to pick up initially missed features. Note that in this process we perform line detection based on a prior, but

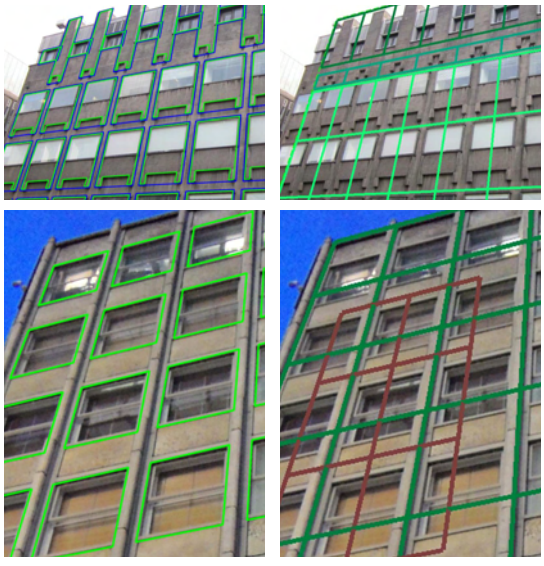


**Figure 6:** *Roughly marking a single repeating element helps to find the correct semantic symmetry (left). Fully automatic methods such as Wu et al. [WFP10] often merge semantically separate parts into one symmetric element (right).*

these priors are learned from the data in the form of symmetry. Thus we look for a base line parameter $l'_1 = (\mathbf{n}, d)$ and a coupling symmetry transform that best explains the observed data. For simplicity we explain the optimization using translations, the most dominant symmetry type in buildings. Translational symmetry encodes the line offset $o$ such that any other line is represented as $l'_i = (\mathbf{n}, d + (i-1)o)$. Let any of the original lines $l_i$ have end points $\mathbf{p}_i^1$ and $\mathbf{p}_i^2$. Then extracting the best line parameters along with the coupling symmetry transform amounts to minimizing

$$E(\mathbf{n},d,o) = \sum_i \|\mathbf{p}_i^1 - \mathbf{p}_i^2\|((\mathbf{n}^T\mathbf{p}_i^1 + d + (i-1)o)^2 + (\mathbf{n}^T\mathbf{p}_i^2 + d + (i-1)o)^2) \quad (2)$$

with the side constraint $\|\mathbf{n}\| = 1$. We alternate between the computation of the transform parameter $o$ and the line parameters $(\mathbf{n}, d)$ using a least squares and an eigen-value formulation, respectively. Once converged (typically 2 to 5 iterations), we recompute the set of close lines to the optimized template strokes and repeat the entire symmetry-based optimization procedure $k = 3$ times. (see Figure 7). The analysis is similar in case of a 2D translational grid. Note that this process is effectively performing symmetrization [MGP07] in the space of lines.

**Structure completion:** So far, we use the projected 3D MVS edges for structure discovery, but left out the original detected edges in each image, since such edges are typically noisy and corrupted with outliers. Now we use the detected regularity among the repeated elements to identify the outlier edges, and make use of the remaining 2D edges. Let $\mathcal{L}^2$ denote the set of image-level edges for any rectified image. Specifically, assuming the detected regularity is a pattern repeated under a 1D or 2D grid structure, we propagate the detected grid structures and also test in regions of missing elements but with reduced threshold margins (75% in our implementation) as compared to the previous stage. Note that instead of projected 3D MVS edges we now make use of edges from $\mathcal{L}^2$. After the missing elements are detected, we reperform the simultaneous line and transformation optimization using all the repeating elements to refine the shape of the repeating elements and the symmetry relation among them. See Figure 7 for a typical example.

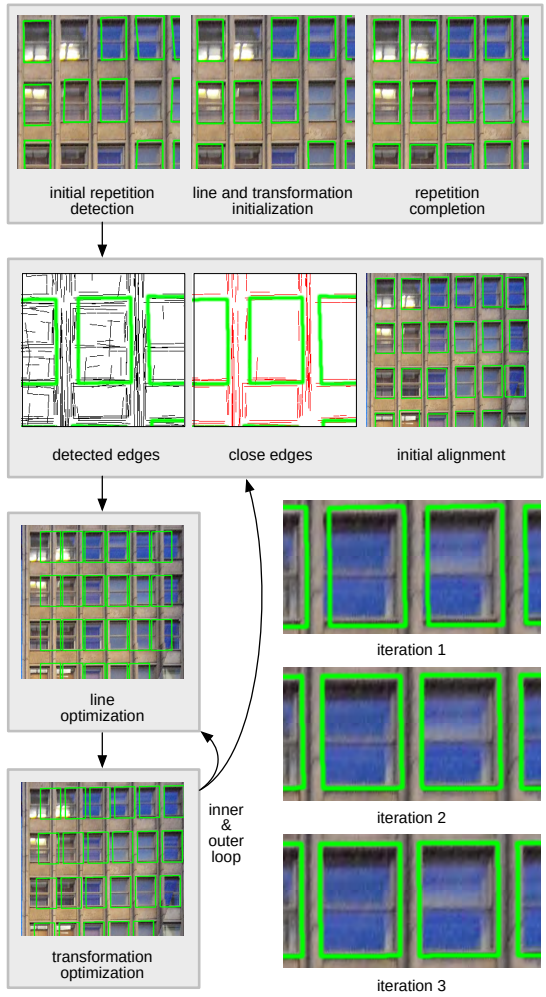


**Figure 7:** *Symmetry based optimization is performed on the initially detected repetitions to initialize the line and the transformation parameters for grid fitting. After the missing elements are detected the optimization procedure is repeated to get the final alignment. This procedure contains an inner loop of successive iterations of line and transformation optimizations and an outer loop of updating the template strokes and reselecting the close edges.*

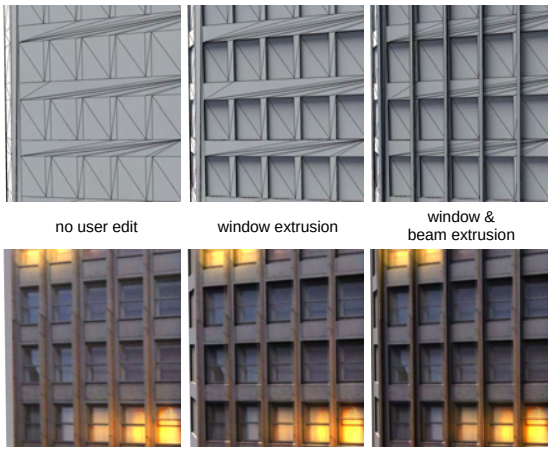


**Figure 8:** *User-guided depth refinement based on the extracted symmetry pattern helps to recover shallow depth features (top: geometry, bottom: textured model).*

**4.4 Procedural Depth Refinement:** At the end of the symmetry-based consolidation we have a set of elements with respective repeating patterns on each of the rectified planes. In practice, however, such patterns typically are offset surfaces from their embedding planes. Hence, we still require depth offsets for such extrusions to produce a 3D model. We recover this depth information in two ways: (i) For each element, we perform a 1D depth search in an offset range of $[-\delta, \delta]$, render the element boundaries using camera parameters of image $I_i$ and compare with the source image $I_i$ based on the 2D-edges $\mathcal{L}^2(I_i)$. In case of insufficient image resolution, however, the method fails to recover shadow depth elements. (ii) Hence we also allow the user to manually prescribe a depth assignment for a *single* element, and the depth is then propagated to all the other symmetrically coupled elements (see Figure 8).

## 5. Evaluation

We evaluate our framework on a variety of challenging real and synthetic scenarios such as non-Lambertian surfaces, abrupt changes in lighting, and small planar patches (see Figures 9 and 10). Table 1 summarizes the performance statistics for the intermediate steps of our approach for each dataset. Please note that these computationally heavy steps are carried out as preprocessing before an interactive reconstruction session begins.

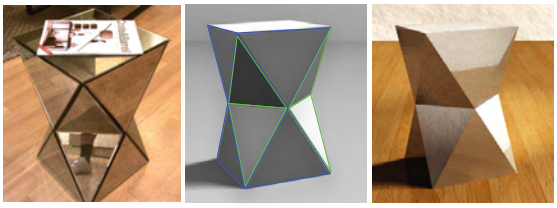


**Figure 9:** *A shiny coffee table with reflective and rotational symmetries reconstructed with our approach. The repeating elements and a realistic rendering are shown on the right.*

We have tested our approach on synthetic examples to measure the accuracy of the symmetry-based optimization step in recovering the correct boundaries of the repeating elements. In Figure 11, we provide a comparison between a ground truth model and our reconstruction obtained by optimizing for the boundaries of the repeating windows and extruding them to the correct depth. We set the maximum distance to 0.5% of the height of the building and provide a color-coded distance measure between the models. We observe small error around the boundaries of the windows and slightly higher error inside the windows due to the depth changes in these regions in the ground truth model. The highest error is produced at the door region where we have missing planes. In this example, the user prescribed only the relevant scale of elements, i.e., windows, and the extrusion depth. Note that the same window element was found and consolidated across multiple planes.

Figure 12 compares our approach with the method of Wu et al. [WFP11] that considers the significantly more challenging scenario of single-view reconstruction. This example illustrates the benefits of our multi-view approach that couples feature and symmetry information across multiple images, leading to more faithful reconstructions in general.

We consider discrete symmetries as multiple observations of the same piece of geometry to reduce noise and perform (moderate) hole filling. Effectively, we integrate information across different symmetric pieces into one consistent representation that is then copied across all instances. We compare this symmetry-aware reconstruction approach with a state-of-the-art MVS algorithm in Figure 13. We provide both the original reconstructed point clouds generated

| | Tower | 3-Sided | SoHo | Mirror | Black | Table | Syn. |
|---|---|---|---|---|---|---|---|
| $\#N_i$ | 25 | 26 | 13 | 9 | 27 | 13 | 24 |
| res | 5.7 | 6.2 | 7.6 | 6.2 | 5.0 | 5.7 | 3.0 |
| $N_e$ | 2600 | 1200 | 2000 | 2400 | 1500 | 750 | 3400 |
| $N_l$ | 2891 | 1570 | 457 | 1128 | 1822 | 173 | 3763 |
| $N_p$ | 2 | 3 | 2 | 6 | 1 | 7 | 5 |
| $N'_p$ | 4 | 0 | 2 | 0 | 0 | 0 | 0 |
| $N'_r$ | 1 | 2 | 1 | 4 | 3 | 2 | 1 |
| $N_r$ | 102 | 80 | 300 | 156 | 57 | 8 | 300 |
| $T_l$ | 55 | 25 | 35 | 40 | 40 | 3 | 45 |
| $T_p$ | 6 | 5 | 4 | 6 | - | 16 | 6 |

**Table 1:** *The table shows the number of input images ($N_i$), the resolution of the images in megapixels (res), the average number of 2D edges detected per image ($N_e$), the number of 3D lines reconstructed ($N_l$), the number of automatically fitted planes ($N_p$), the number of manually selected planes ($N'_p$), the number of elements marked by the user ($N'_r$), and the total number of repeating elements detected ($N_r$) for each data set. The computation times for 3D line reconstruction ($T_l$) and plane-based image segmentation ($T_p$) are given in minutes measured on a 3.33 MHz 24-core machine.*

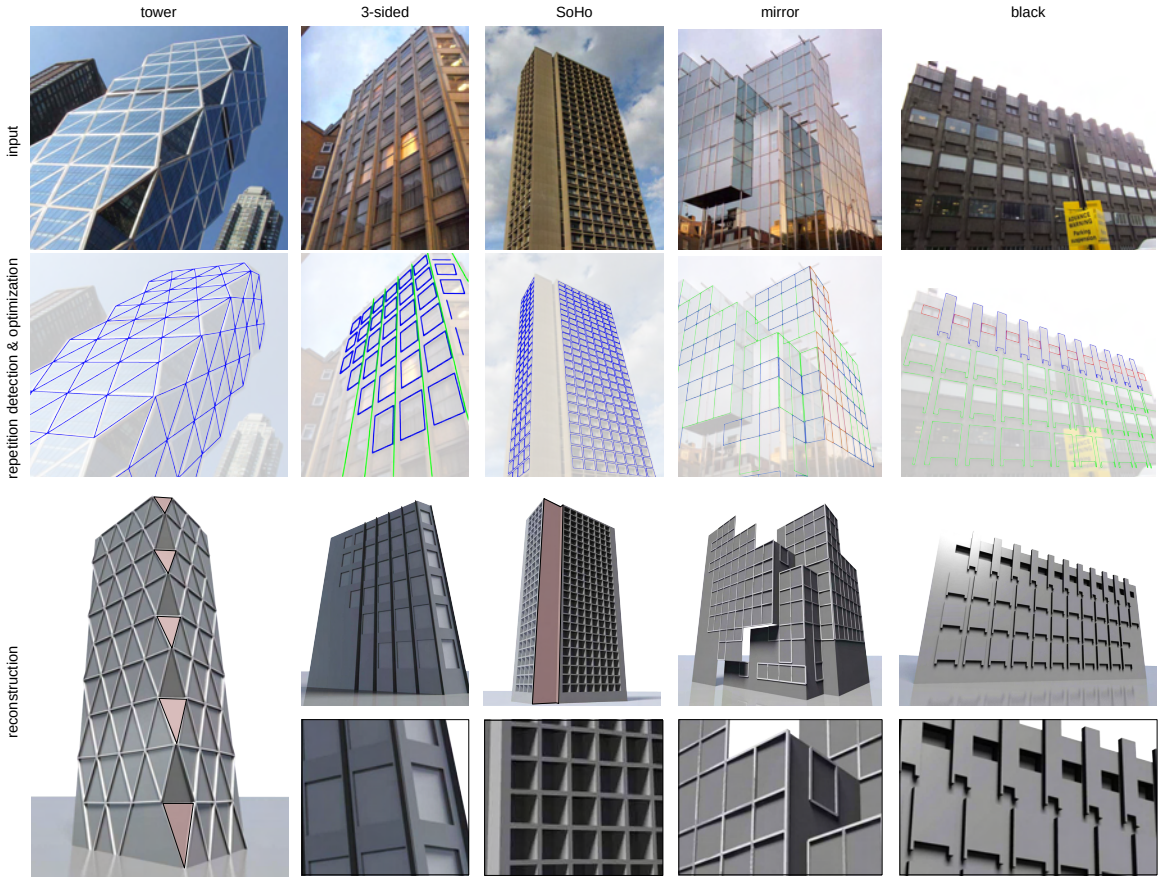


**Figure 10:** *Urban buildings with complex symmetries and non-trivial textures. The middle row shows the optimized repetition patterns with different colors indicating separate structures. The final reconstructions are shown in the bottom row, where the red planes have been added with user assistance due to a lack of stable line features.*

by the MVS algorithm and surfaces obtained by Poisson Surface Reconstruction (PSR) [KBH06]. The MVS algorithm produces noisy and sparse point sets, especially for non-Lambertian surfaces, and PSR creates a smooth surface while filling in the holes with blobs. To our knowledge, no competing method can robustly handle such challenging scenarios. In contrast, our initial edge-based stereo approach enables to distinguish between the spurious features due to reflections and the actual features and initializes a consistent reconstruction that preserves the sharpness. By incorparating this distinction into the symmetry detection process, information is also propagated across the (detected) repeating elements. Additionally, we obtain a compressed representation that enables not only efficient data storage, but can directly be used for structure-aware edits of the geometry.

Our 2D-3D coupled repetition detection algorithm uses a weighted score of image-based normalized cross-correlation (NCC) score and line-based similarity to compare elements. For examples where there are sufficient image features, e.g. black, SoHo in Figure 10, NCC matching provides a good initialization of the present regularity. On the other hand, as the surfaces become more reflective and textureless, e.g mirror and table datasets, image-based comparisons become inaccurate, while 3D linear features provide a more reliable result. Hence, we normally use an equally weighted combination of image- and line-based similarity measures but rely only on line-based similarity for highly reflective surfaces to initialize our regularity discovery. The symmetry-based optimization aids the initialization and helps to discover the remaining missing repeating elements, which are otherwise challenging to detect.

**User interactions:** We support three types of user interactions: (i) After the automatic computation of the plane-based image segmentation, there might be mislabeled regions or missing planes due to insufficient 3D lines, especially for regions with little support (e.g., thin planes). In both cases, the user can indicate rough strokes on the images that get snapped to the 2D edges either to define new segment boundaries or to fit new planes. In order to fit new planes, we require the user to mark two edges to define the plane in two images which are converted to 3D lines to compute the plane parameters in 3D. Finally, intersections with

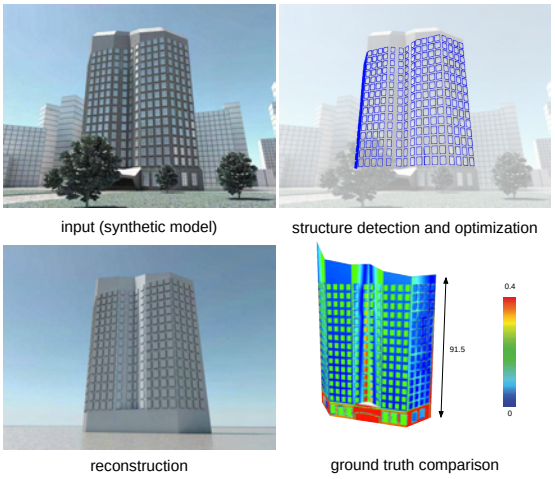


**Figure 11:** *Computations with a synthetically rendered building demonstrates the accuracy of our method.*

the current planes are used to define the boundaries of the new plane. In our examples, this mode was only necessary in the mirror dataset to define new plane boundaries (see Figure 4) and for the tower and the Soho datasets to indicate additional planes (see pink regions in Figure 10). The 3D line densities were sufficient for the other examples for the algorithm to correctly fit 3D planes. Note that the 3-sided building or the table contain non-axis aligned planes, which do not satisfy the Manhattan-world assumption [FCSS09a]. (ii) Often there is an ambiguity between semantically correct element boundaries and the scale of the repeating elements that is difficult to resolve automatically (see Figure 6). For humans, however, it is trivial to roughly mark a representative element of the intended regularity. Therefore we allow users to roughly indicate a single element, which is then used to detect the other repetition instances. As seen in Table 1, with only a few user marked elements our algorithm can detect almost a complete set of repetitions. (iii) We also allow the user to indicate shallow extruded feature depth similar to Müller et al. [MZWG07] (see Figure 8).

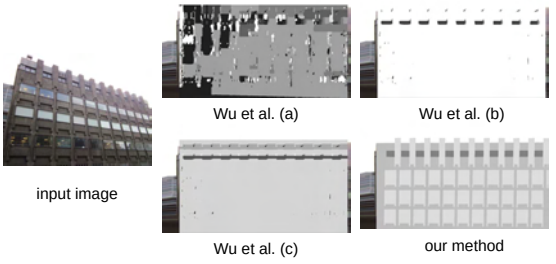


**Figure 12:** *The method of [WFP11] fails to recover the depth of the repeating elements if the depth change with respect to the main plane is too small. Depth assignments obtained by different weighting of the repetition and smoothness terms are provided ((a) no repetition term, (b) repetition and smoothness terms weighted equally, (c) smoothness term weighted more) in comparison to our method.*

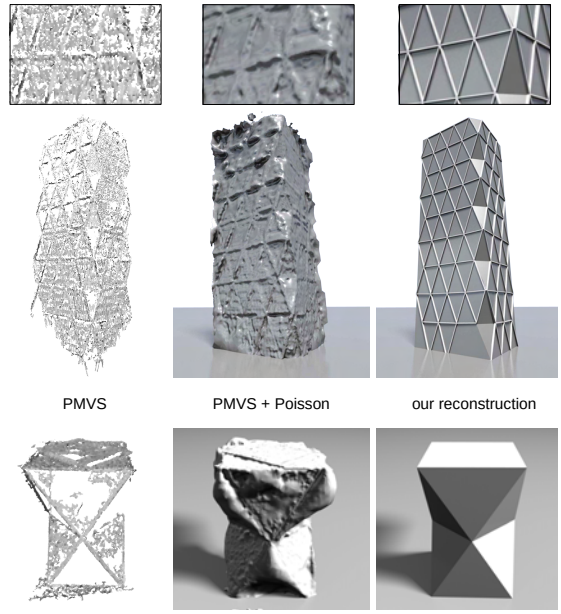


**Figure 13:** *The comparison with the patch-based MVS method of [FP09] illustrates that symmetry priors and non-local consolidation are essential for the objects with complex materials and repetition patterns (see supplementary).*

**Limitations:** Even when exploiting symmetry priors, surface reconstruction from images remains an ill-posed problem, hence our method will fail if 2D edge detection, 3D line estimation, or 3D candidate plane computation do not provide sufficient geometric information. Similarly, symmetry detection will be ineffective in cases of limited repetition or strong variations in the repeating elements (e.g. due to weathering). Our current pipeline focuses on piecewise planar surfaces bounded by straight edges, as are mostly common in modern urban buildings. Curved edges or surfaces are currently not handled by our method. Integrating such features offers interesting opportunities for future work.

## 6. Conclusions and Future Work

We presented a coupled formulation for detecting symmetric line arrangements and 3D reconstruction for producing *factored facade* models. Unlike most competing approaches, we benefit from large-scale model repetitions, and can robustly handle inputs with strong reflections, shadow elements, or outlier objects, which are impossible to disambiguate with only local reasoning. We bootstrap the reconstruction using rough image-space user markings, and subsequently use the factored facades to allow symmetry coupled non-local 3D edits. In the future, we plan to handle not just buildings, but also large city blocks and building

colonies where large-scale repetitions are abundant. In our current formulation, we assumed the initial camera calibration to be fixed — we plan to refine the calibration using a generalized formulation coupling calibration, symmetry detection, and 3D modeling.

**Acknowledgements:** This research has been supported by the ERC Starting Grant 257453 COSYM, a KAUST visiting student grant, and the Marie Curie Career Integration Grant 303541.

## References

[AFS*10] AGARWAL S., FURUKAWA Y., SNAVELY N., CURLESS B., SEITZ S. M., SZELISKI R.: Reconstructing Rome. *IEEE Computer* (2010), 40–47. 1

[BBW*09] BOKELOH M., BERNER A., WAND M., SEIDEL H.-P., SCHILLING A.: Symmetry detection using line features. *Computer Graphics Forum 28* (2009), 697–706. 3

[CKX*08] CHEN X., KANG S. B., XU Y.-Q., DORSEY J., SHUM H.-Y.: Sketching reality: Realistic interpretation of architectural designs. *ACM TOG 27* (2008), 11:1–11:15. 3

[DS08] DU SAUTOY M.: *Symmetry: A Journey into the Patterns of Nature*. Harper, 2008. 5

[DTM96] DEBEVEC P. E., TAYLOR C. J., MALIK J.: Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Proc. of SIGGRAPH* (1996). 3

[FCSS09a] FURUKAWA Y., CURLESS B., SEITZ S., SZELISKI R.: Manhattan-world stereo. In *CVPR* (2009). 2, 4, 9

[FCSS09b] FURUKAWA Y., CURLESS B., SEITZ S. M., SZELISKI R.: Reconstructing building interiors from images. In *ICCV* (2009). 2

[FJZ05] FRUEH C., JAIN S., ZAKHOR A.: Data processing algorithms for generating texture 3D building facade meshes from laser scans and camera images. *IJCV 61* (2005), 159–184. 3

[FP09] FURUKAWA Y., PONCE J.: Accurate, dense, and robust multiview stereopsis. *IEEE PAMI 32* (2009), 1362–1376. 1, 2, 9

[GSC*07] GOESELE M., SNAVELY N., CURLESS B., HOPPE H., SEITZ S. M.: Multi-view stereo for community photo collections. In *ICCV* (2007). 2

[JTC09] JIANG N., TAN P., CHEONG L.-F.: Symmetric architecture modeling with a single image. *ACM TOG 28* (2009), 113:1–113:8. 3

[JTC11] JIANG N., TAN P., CHEONG L.-F.: Multi-view repetitive structure detection. In *IEEE International Conference on Computer Vision (ICCV)* (Barcelona, Spain, 2011). 5

[KBH06] KAZHDAN M., BOLITHO M., HOPPE H.: Poisson surface reconstruction. In *Symp. on Geometry Processing* (2006). 8

[Kol06] KOLMOGOROV V.: Convergent tree-reweighted message passing for energy minimization. *IEEE PAMI 28* (2006), 1568–1583. 4

[KW11] KELLY T., WONKA P.: Interactive architectural modeling with procedural extrusions. *ACM TOG 30* (2011), 14:1–14:15. 2

[KZ02] KOŠECKÁ J., ZHANG W.: Video compass. In *ECCV* (2002), Springer-Verlag, pp. 657–673. 2

[LZS*11] LI Y., ZHENG Q., SHARF A., COHEN-OR D., CHEN B., MITRA N. J.: 2d-3d fusion for layer decomposition of urban facades. In *IEEE International Conference on Computer Vision (ICCV)* (Barcelona, Spain, 2011). 3, 5

[MGP07] MITRA N. J., GUIBAS L., PAULY M.: Symmetrization. *ACM TOG 26*, 3 (2007), 63:1–63:8. 6

[MK10] MIČUŠÍK B., KOŠECKÁ J.: Multi-view superpixel stereo in urban environments. *IJCV 89* (2010), 106–119. 2

[MWR*09] MUSIALSKI P., WONKA P., RESHEIS M., MAIERHOFER S., PURGATHOFER W.: Symmetry-based facade repair. *Vision. Modeling, and Visualization* (2009). 3

[MZWG07] MÜLLER P., ZENG G., WONKA P., GOOL L. V.: Image-based procedural modeling of facades. *ACM TOG 26*, 3 (2007). 3, 9

[NSZ*10] NAN L., SHARF A., ZHANG H., COHEN-OR D., CHEN B.: Smartboxes for interactive urban reconstruction. *ACM TOG 29*, 4 (2010), 93:1–93:10. 3

[PMW*08] PAULY M., MITRA N. J., WALLNER J., POTTMANN H., GUIBAS L.: Discovering structural regularity in 3D geometry. *ACM TOG 27*, 3 (2008), 43:1–43:11. 3, 5

[SCD*06] SEITZ S. M., CURLESS B., DIEBEL J., SCHARSTEIN D., SZELISKI R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR* (2006). 2

[SJW*11] SUNKEL M., JANSEN S., WAND M., EISEMANN E., SEIDEL H.-P.: Learning line features in 3d geometry. *Computer Graphics Forum 30* (2011), 267–276. 3

[SSS06] SNAVELY N., SEITZ S. M., SZELISKI R.: Photo tourism: exploring photo collections in 3d. *ACM TOG 25* (2006), 835–846. 4

[SSS*08] SINHA S. N., STEEDLY D., SZELISKI R., AGRAWALA M., POLLEFEYS M.: Interactive 3d architectural modeling from unordered photo collections. *ACM TOG 27* (2008), 159:1–159:10. 3

[SSS09] SINHA S. N., STEEDLY D., SZELISKI R.: Piecewise planar stereo for image-based rendering. In *Computer Vision, 2009 IEEE 12th International Conference on* (29 2009-oct. 2 2009), pp. 1881 –1888. 4

[SZ97] SCHMID C., ZISSERMAN A.: Automatic line matching across views. In *CVPR* (1997). 2

[VAW*10] VANEGAS C., ALIAGA D., WONKA P., MÜLLER P., WADDELL P., WATSON B.: Modeling the appearance and behavior of urban spaces. *Computer Graphics Forum 29*, 1 (2010), 25–42. 1

[WFP10] WU C., FRAHM J.-M., POLLEFEYS M.: Detecting large repetitive structures with salient boundaries. In *ECCV* (2010). 5, 6

[WFP11] WU C., FRAHM J.-M., POLLEFEYS M.: Repetition-based dense single-view reconstruction. In *CVPR* (2011). 3, 7, 9

[WWSR03] WONKA P., WIMMER M., SILLION F., RIBARSKY W.: Instant architecture. *ACM TOG 22* (2003), 669–677. 2

[WZ02] WERNER T., ZISSERMAN A.: New techniques for automated architecture reconstruction from photographs. In *ECCV* (2002). 2

[XFT*08] XIAO J., FANG T., TAN P., ZHAO P., OFEK E., QUAN L.: Image-based façade modeling. *ACM TOG 27* (2008), 161:1–161:10. 3

[XFZ*09] XIAO J., FANG T., ZHAO P., LHUILLIER M., QUAN L.: Image-based street-side city modeling. *ACM TOG 28* (2009), 114:1–114:12. 3

[ZSW*10] ZHENG Q., SHARF A., WAN G., LI Y., MITRA N. J., COHEN-OR D., CHEN B.: Non-local scan consolidation for 3D urban scenes. *ACM TOG 29*, 4 (2010), 94:1–94:9. 3