Coupled Structure-from-Motion and 3D Symmetry Detection for Urban Facades

DUYGU CEYLAN
EPFL
NILOY J. MITRA
University College London
YOUYI ZHENG
KAUST
and
MARK PAULY
EPFL

Repeated structures are ubiquitous in urban facades. Such repetitions lead to ambiguity in establishing correspondences across sets of unordered images. A decoupled structure-from-motion reconstruction followed by symmetry detection often produces errors: outputs are either noisy and incomplete, or even worse, appear to be valid but actually have a wrong number of repeated elements. We present an optimization framework for extracting repeated elements in images of urban facades, while simultaneously calibrating the input images and recovering the 3D scene geometry using a graph-based global analysis. We evaluate the robustness of the proposed scheme on a range of challenging examples containing widespread repetitions and nondistinctive features. These image sets are common but cannot be handled well with state-of-the-art methods. We show that the recovered symmetry information along with the 3D geometry enables a range of novel image editing operations that maintain consistency across the images.

Categories and Subject Descriptors: I.4.1 [**Image Processing and Computer Vision**]: Digitization and Image Capture—*Camera calibration*; I.4.5 [**Image Processing and Computer Vision**]: Reconstruction; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Shape*

General Terms: Algorithms, Design

Additional Key Words and Phrases: Structure-from-motion, symmetry detection, image collections, 3D reconstruction, coupled editing

ACM Reference Format:

Duygu Ceylan, Niloy J. Mitra, Youyi Zheng, and Mark Pauly. 2014. Coupled structure-from-motion and 3D symmetry detection for urban facades. ACM Trans. Graph. 33, 1. Article 2 (January 2014) 15 pages.
DOI: http://dx.doi.org/10.1145/2517348

1. INTRODUCTION

Acquiring 3D geometry of urban spaces is central to many applications for digital cities, including mapping and navigation, urban design, and content creation for entertainment. Such applications heavily depend on accurate 3D models of building facades since many tasks require interaction with street-level buildings, for example, in Google Street View or Microsoft Visual Earth. While different 3D acquisition possibilities exist, image-based modeling remains the most popular one due to the simplicity of image acquisition and the growing ubiquity of handheld cameras (see Quan and Kanade [2010] for a detailed history of image-based modeling).

Any such image-based modeling method relies heavily on accurate calibration of the scene, that is, the location, orientation, and field-of-view of the camera corresponding to each image, to determine the 3D scene geometry by triangulating the calibrated images [Snavely et al. 2006]. The core challenge is to establish correspondence, that is, identify points across images that relate to the same 3D position in the physical world, while factoring out image space variations due to occlusion, reflection, zoom level, etc. The problem becomes particularly difficult in the presence of repeated elements that give rise to multiple and ambiguous correspondences as in the case of urban facades, where symmetry often forms an organizing principle due to aesthetic considerations, manufacturing efficiency, and economic implications. Unfortunately, such widespread repetitions make stable correspondence estimation difficult, potentially leading to poor reconstruction results.

Traditional image-based methods that do not explicitly take symmetries into account often contain one of the following artifacts:

This research received funding from the European Research Council under the European Unions 7th Framework Programme (FP/2007-2013)/ERC Grant Agreement 257453, ERC Starting Grant COSYM and European Community's Research Infrastructure Action grant agreement VISIONAIR 262044 under the 7th Framework Programme (FP7/2007-2013). This research was also supported by a Marie Curie CIG, an Adobe Research grant, and a UCL impact award.

Authors' addresses: D. Ceylan (corresponding author), EPFL, Route Cantonale, 1015 Lausanne, Switzerland; email: duygu.ceylan@epfl.ch; N. J. Mitra, University College London, Gower St, London WC1E 6BT, UK; Y. Zheng, KAUST, Thuwal Saudi Arabia; M. Pauly, EPFL, Route Cantonale, 1015 Lausanne, Switzerland.

DOI: http://dx.doi.org/10.1145/2517348

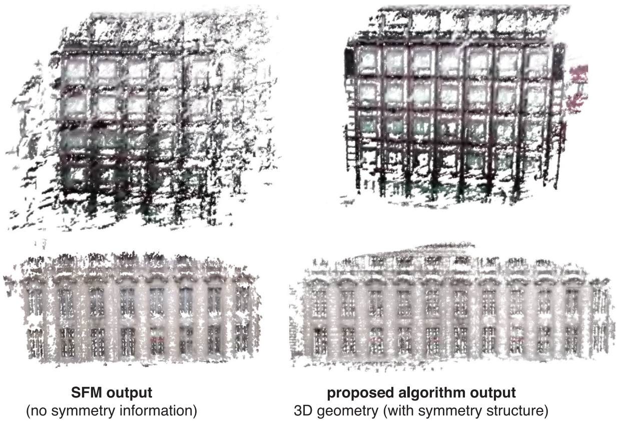
SFM output
(no symmetry information)
proposed algorithm output
3D geometry (with symmetry structure)

Fig. 1. Repetitions in urban facades result in ambiguous correspondences across image pairs that cause traditional Structure-from-Motion (SfM) to fail: Either reconstruction quality is poor, or the reconstruction looks apparently plausible but actually has the wrong number of elements. In contrast, our coupled regularity detection and image calibration framework recovers accurate 3D facade geometry along with the repetition structure.

(i) large-scale ambiguities due to content repetition cause standard Structure-from-Motion (SfM) to produce poor and noisy 3D output, or (ii) SfM produces apparently reasonable 3D output, but with an incorrect number of repeated elements (see Figure 1). Even if the calibration is seemingly successful, the results can be suboptimal, producing sparse, incomplete 3D reconstructions that often accumulate error leading to drifts (e.g., straight facades appearing curved) (see Figure 13).

We propose an image-based 3D reconstruction and editing framework that avoids these difficulties by injecting symmetry information early into the reconstruction process. Our algorithm detects regularities and establishes correspondence simultaneously, thereby addressing the cyclic dependency of the problem of 3D reconstruction with repeating structures: stable 3D symmetry detection requires accurate camera calibration to obtain correct 3D point samples, while accurate camera calibration requires stable symmetry detection to resolve ambiguities.

We bootstrap the algorithm by first detecting regularities explicitly in the 2D images and then use the symmetry information to guide the correspondence search. We perform regularity detection based on a user-marked template in a single image (the only user assistance in our algorithm). We incorporate geometric relations across the detected repeating structures as constraints in a novel graph-based optimization. This optimization simultaneously disambiguates correspondences while estimating camera poses and refining the symmetry relations. As output, we obtain a globally consistent 3D geometry reconstruction with explicit encoding of the facade regularities. In Figure 2, we show a typical reconstruction result, along with the recovered symmetry grid. For this example, state-of-the-art SfM approaches or even a method that tries to detect ambiguous relations [Zach et al. 2010] fail to produce any reasonable output (see supplementary material for details).

In this article we focus on repetitions arranged as planar grids. These are the dominant type of regularities in building facades [Pauly et al. 2008]. This restriction significantly reduces the search space of geometric relations across images since any transformation induced by correspondences between images should overlap the boundaries of the repeating elements. *Consistently* ordering all the images by considering only their pairwise relations is, however, unlikely to be successful. Instead, we consider collections of pairwise relations and present a global optimization scheme to resolve ambiguities. We further optimize the solution by refining the initial vanishing line estimates and perform another pass of constrained bundle adjustment to finalize the camera parameters and the extracted 3D grid parameters.

We evaluate the accuracy and robustness of our framework on a large number of challenging datasets with single and multiple facades, where traditional SfM approaches either fail or perform poorly. In addition to accurate reconstruction, our algorithm also extracts regularity information and effectively couples the 2D input images with the reconstructed 3D output. We use this information to enable coupled image manipulations that are otherwise difficult to achieve.

Contributions. Our key contribution is to couple camera calibration with symmetry detection in the context of urban facade reconstruction. Such a coupling has the following advantages.

First, explicit detection of the repeating structures reduces the search space of possible geometric relations between image pairs and globally disambiguates the correspondence search. Second, exposing explicit constraints about the geometric relations between the repeating structures significantly increases the reconstruction accuracy. Finally, the computed 3D scene geometry along with the extracted regularity links the input images and the reconstructed 3D scene, which in turn enables several interactive editing applications.

2. RELATED WORK

Structure-from-Motion (SfM). A multitude of SfM algorithms have been developed in recent years to compute camera parameters and sparse 3D scene geometry from image sequences. Typically, such algorithms work in two key stages. In the first stage, overlapping images are identified and correspondences across different views are established. In the second stage, the extracted correspondences are used to geometrically relate the views and estimate both the camera parameters and the 3D structure.

An important category of SfM algorithms operate on ordered image sequences such as video [Gil et al. 2006; Pollefeys et al. 2008]. The known order of the images provides strong cues to determine candidate images that should be matched.

In another approach, the growing availability of large numbers of images of particular places from the Internet has inspired researchers to develop SfM methods for unorganized image sets (see Snavely [2011] and references therein). However, without any assumption on the order of the images, matching images and establishing their global relations is challenging, especially in the presence of visually similar and repeating structures. Several approaches have been proposed: Govindu [2004] uses the redundancy in the pairwise image relations to average multiple observations to produce a globally consistent motion estimation, and later [Govindu 2006] randomly samples spanning trees from a graph encoding image relations to prune out mismatches. Martinec and Pajdla [2007] incrementally remove high-residual matches to increase robustness; Zach et al. [2008] use features that are matched between two images but not detected in a third view to detect false matches; while Klopschitz et al. [2010] propose an incremental framework that favors subsets of images with the highest local connectivity. Micusik et al. [2008] use rectangular structures in two-view matching as an alternative to ambiguous feature points. Such *local methods*, however, can fail to produce high-quality outputs in the presence of widespread nonlocal ambiguity.

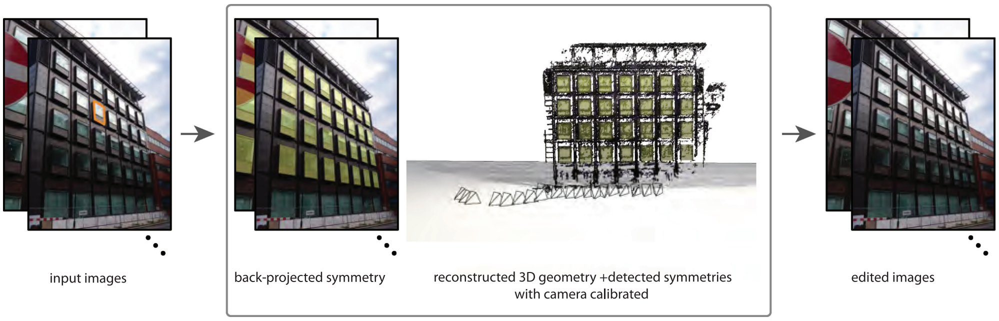

Fig. 2. Starting from a set of 26 unordered images of a facade (left), we propose a global optimization framework that simultaneously recovers 3D scene information, extracts 2-parameter repetition patterns, and calibrates the input images (middle). The user marks one repeated element in one of the images (left). The 3D geometry along with the repetition information can then be used for novel image manipulations, for example, changing the repetition patterns (right), or removing unwanted image elements, consistently across all the input images.

Analyzing image collections. In the context of global analysis, Heath et al. [2010] study graph connectivity across massive image collections to identify interesting visual pathways, while Zach et al. [2010] use a graph to encode visual relations in image collections and infer false matches based on inconsistencies of cycles in this graph. However, they do not explicitly model the repetitions as in our method, which we found to improve the quality of the results significantly. In a follow-up work, Cohen et al. [2012] propose to use symmetry priors with collinearity and orthogonality constraints in an additional bundle adjustment step to reduce drifts in a given initial SfM output. They use the method of Zach et al. [2010] to compute the initial SfM output which is assumed free of ambiguity. In contrast, we jointly resolve ambiguities and enforce symmetry constraints in SfM. A similar approach based on loop consistency has been recently applied to improve collections of maps across 3D shapes [Nguyen et al. 2011]. Roberts et al. [2011] focus on a specific instance of the image matching problem where large duplicate structures are present in the scene. They build a probabilistic framework in which nongeometric cues such as the image timestamps are combined with geometric ones. In a recent effort, Jiang et al. [2012] also propose a framework to handle ambiguities with a special focus on scenes with large duplicate structures. They eliminate the dependency on the image timestamps by formulating the problem as finding the spanning tree of a matching graph minimizing a global energy function and propose a greedy search algorithm. Our approach is inspired by these methods to establish globally consistent image relations. Unlike all other previous efforts, however, we *jointly* focus on regularity detection and constrained SfM formulation. By explicitly detecting regular structures (i.e., planar 2-parameter grids) as an initial step, we significantly reduce the search space for finding the optimal transformation that relates a pair of images. We show that by simultaneously calibrating the cameras, extracting the 3D regularities, and reconstructing the 3D geometry, we achieve significant improvements both in terms of robustness and accuracy.

Structural priors. Structural priors, when available as in the case of urban scenes, have been used to improve the output quality of image-based reconstruction methods (see Musialski et al. [2012] and references therein). For example, coplanarity constraints [Bartoli and Sturm 2003] and vanishing points [Sinha et al. 2010] have been explored in several SfM methods. Xiao et al. [2008] use SfM output to represent building facades as rectangular or developable surfaces. Recently, Wan et al. [2012] propose a framework to reconstruct piecewise-planar buildings that incorporates constraints based on the relations between the facades of a building. Such methods, however, *assume* access to the detected structures.

A potential solution is to first extract 3D structures from initial SfM reconstructions and then use the constraints to regularize the output. Such a decoupled approach, however, can fail since the low-quality 3D geometry makes robust structure detection difficult. In our tests even with high-resolution inputs, the output point sets from Bundler [Snavely et al. 2006] or Zach et al. [2010] are too coarse to reliably initialize transform domain grid fitting as proposed by Pauly et al. [2008]. While such grid extraction methods can succeed for clean and dense SfM output [Jiang et al. 2011], the results of such a decoupled approach degrade sharply when the input images are of low resolution or have significant ambiguities (see Figure 14). Intuitively, the lack of reliable feature detectors for sparse point sets and accumulation errors in the initial reconstruction lead to poor initial grid estimates, which cannot be corrected by subsequent local refinements. Further, if the SfM outputs have gross errors due to ambiguity across repeated elements (see Figure 1), subsequent analysis of the 3D point sets can only extract wrong constraints. Other possibilities involve integrating information from other acquisition modes (e.g., LiDAR scans as used by Zheng et al. [2010] and Li et al. [2011]) or allowing the user to indicate symmetry structures on 3D data (e.g., Nan et al. [2010]) (see the recent survey of Mitra et al. [2013] for a more detailed discussion on the use of symmetry priors in architecture modeling). In contrast, we demonstrate that even a purely image-based symmetry detection and scene reconstruction can successfully produce high-quality 3D geometry without relying on specialized acquisition setups (e.g., LiDAR and aerial scans) or extensive manual work.

Symmetry for image-based modeling. Symmetry can be exploited to extract dense 3D reconstruction even from a single image as shown by Wu et al. [2011]. They extract image-level horizontal repetitions to establish pixel-level correspondences for camera estimation and 3D reconstruction. Earlier, Jiang et al. [2009] perform camera calibration from a single image by exploiting symmetry and produce textured polygonal reconstructions guided by interactive user annotations. These methods, however, assume the complete facade to be visible in a single image. Such assumptions are often violated as in all our test image sets.

Ceylan et al. [2012] propose a method to accurately reconstruct urban buildings from multiple images using lines as prior. However,

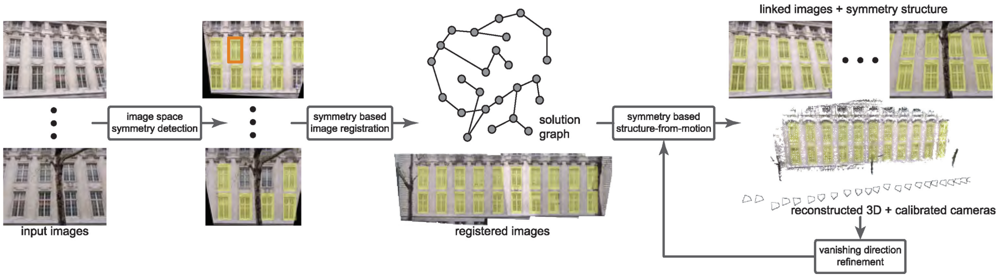
linked images + symmetry structure
image space symmetry detection
symmetry based image registration
solution graph
symmetry based structure-from-motion
input images
registered images
reconstructed 3D + calibrated cameras
vanishing direction refinement

Fig. 3. Starting from a set of unordered facade images (25 images in this example), we first rectify the individual images. We then ñpropagate the user-marked repeated element (shown in orange) in one image to all the other images and perform symmetry detection in each image. We use the initial repetition information (shown in yellow) to solve for a consistent global repetition pattern using a graph-based optimization. We perform a symmetry-based SfM reconstruction to simultaneously calibrate the cameras and extract a 3D reconstruction. With the obtained 3D scene structure, we refine the initial rectification and repeat the SFM computation. The reconstruction along with the resultant back-projected repeated elements on the images can then be used to enable a range of editing possibilities.

the method assumes the input images to be precalibrated, which is difficult in the presence of repetition ambiguities. In contrast, we simultaneously extract facade repetitions and scene calibration leading to reliable 3D reconstruction. Note that, different from all other methods, we directly solve for symmetry coupling that is then used for nonlocal image edits (see Section 6).

3. OVERVIEW

Figure 3 shows an overview of our processing framework (see also Algorithm 1). Our algorithm takes as input a set of unordered images $\mathcal{I} := \{I_1, \ldots, I_n\}$ of a building, which we assume to contain one or multiple 1- or 2-parameter repetition patterns on each facade. (In the absence of such repetitions, our algorithm produces standard SfM output.) Note that neither the full facade repetition grid needs to be completely visible in any single image, nor do the repetitions have to be restricted to a single facade.

After the images have been rectified, the user roughly indicates elements of interest, for example, a repeating window frame visible across the input images, on any *one* of the input images. This is the only manual assistance required in our reconstruction pipeline. We then compute initial estimates for the grid generators of the repetition patterns (Section 4.1).

Next, in a key algorithmic stage, we pose correspondence search across images in $\mathcal{I}$ as solving for offset positions for each image on an (unknown) global repetition grid (Section 4.2). We simultaneously solve for offset positions and estimate the associated camera parameters for each image, while also detecting the 3D repetition pattern. The optimization is formulated as selecting a subset of consistent edges from a *matching graph*, where a node corresponds to an image in $\mathcal{I}$ and each edge denotes an estimated image-pair alignment, which can possibly be wrong. We solve for a consistent set of edge alignments by progressively refining the alignments.

Subsequently, we perform bundle adjustment with the extracted symmetries as constraints that encode the grid arrangements of the repeating structures (Section 4.3). We iterate by refining the initial vanishing line estimates (Section 4.4) and also extend our algorithm to handle multiple repetition grids (Section 4.5). Thus our algorithm outputs the final camera poses and a sparse 3D reconstruction of the scene together with the refined symmetry parameters in 3D. We then use this information to explore a range of editing possibilities coupled across multiple images with the detected symmetries providing nonlocal binding.

4. ALGORITHM

In this section, we provide the details of the individual components of our pipeline (see Algorithm 1).

ALGORITHM 1: Algorithm pseudocode (Section 4).

```
Input: A set of images $I$ and a user marked template $T$
Output: Camera parameters $\forall I_i \in \mathcal{I}$, 3D scene structure,
        and the 3D symmetry information
foreach $I_i \in I$ do
    $I_i \leftarrow$ Rectify($I_i$);
    InitialGridEstimation($I_i, T$);
end
/* Symmetry-based Image Registration */
define $G := (V, E)$, $V \leftarrow \mathcal{I}$, $E \leftarrow \emptyset$;
foreach $I_i, I_j \in \mathcal{I}$ do
    if FindCandidateAlignment($I_i, I_j$) then
        $E \leftarrow e_{ij} \cup E$;
    end
end
$\{\chi_e\} \leftarrow$ ImageMatchingOptimization($E$);
$M \leftarrow$ FindMinimumSpanningTree($G, \{\chi_e\}$);
SymmetryBasedSFM($M$);
VanishingLineRefinement();
SymmetryBasedSFM($M$);
```

4.1 Initial Grid Estimation

Our goal is to bring the input facade images $\mathcal{I}$ into correspondence. We exploit the underlying facade repetition structure to restrict the continuous space of alignments to a discrete set of possibilities, that is, correspondence across an underlying 2-parameter repetition grid. Hence, in this preprocessing stage, we detect such repetition grids for each image, which are then used in Section 4.2 to globally establish correspondence across all the individually extracted grids. We start by analyzing each image as described next.

Building facades typically contain dominant repetitions arranged along vertical and horizontal directions. Therefore, we first detect dominant vanishing points in each image $I_i \in \mathcal{I}$ using the cascaded

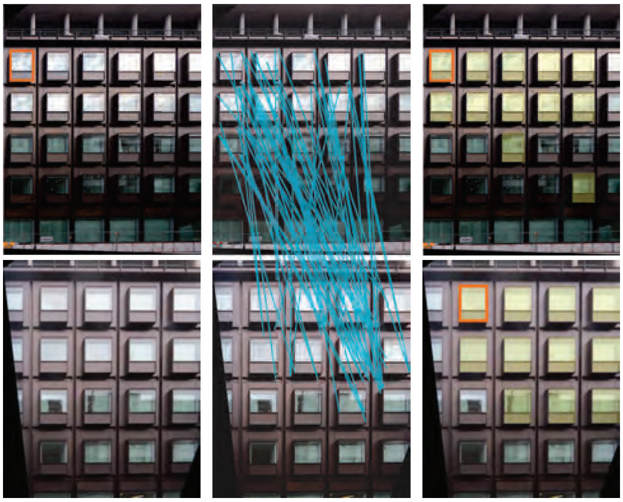

Fig. 4. Given a pair of rectified images (left) with a repeating element T marked in one image (top-left), we use SIFT features to estimate the scale factor s relating the image pair (middle). We then create a scaled template sT suitable for the other image. Subsequently, we detect image-level repetitions (marked in yellow) across each individual rectified image (right).

Hough transform [Tuytelaars et al. 1998] and use these vanishing points to rectify the original images to be fronto-parallel. (With slight abuse of notation, we use I_i to also denote the corresponding rectified images.) Note that for images where multiple facades are visible, multiple vanishing points are detected and rectification is performed with respect to *each* of these candidate vanishing points. In the successive stages, we detect repetitions in each of these rectified images which may result in multiple detected grids for a single image.

Since the rectified images can be at different scales, we match features across them to estimate their relative scalings. Specifically, since rotation effects are compensated through rectification, we extract SIFT features with fixed upright orientation and estimate the scale change between a pair of rectified images by clustering the scale differences between the feature matches (see also Baatz et al. [2010]) (see Figure 4). Note that ambiguous feature matches arising from repetitions *do not* affect the scale estimation as SIFT features detected in a rectified image have similar scales across the repeating elements. Thus, at the end of this step, we have an estimated scale s_i for each image I_i. We set $s_1 = 1$.

User interaction. In order to prescribe the element of interest, the user marks a single repeated element T_1 in any one of the rectified images, say I_1. Then for all other images I_i, we scale the element using the estimated scales s_i, that is, $T_i \leftarrow s_i T_1$ and use the scaled element as template for the respective image (see Figure 4). We perform template matching in the images by comparing local image patches based on the (scaled) template regions using Normalized Cross-Correlation (NCC) (see also Ceylan et al. [2012]). Note that although other image-based symmetry detection methods (e.g., [Wu et al. 2010a]) can also be used in this step to automatically select a template, we found it desirable to allow the user to indicate the template in a single image to ensure we work on an element that is visible across multiple images. Furthermore, this user assistance identifies a semantically meaningful repeating element which can be used in a variety of editing applications.

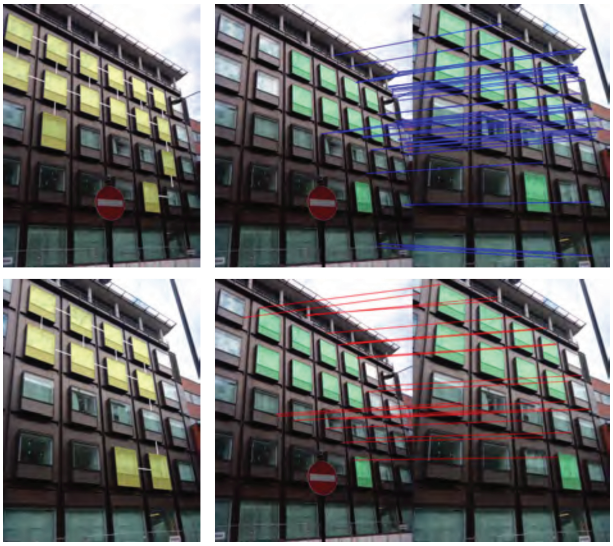

Fig. 5. Symmetry guided feature matching: Given a pair of images with their detected grids (left), for each candidate alignment we detect the overlapping grid regions (shown in green). We compute feature matches outside the overlapping regions and count the number of matches that support the candidate alignment. Top-right shows the correct alignment with highest support (51 supporting feature matches are in blue) and the bottom-right is a wrong candidate alignment (25 supporting feature matches are in red).

Finally, for each image, we complete the initialization by fitting a grid to the detected elements and estimating its corresponding grid generators [Pauly et al. 2008] (see Figure 5 (left)). Note that missing grid elements are recovered in the subsequent stages.

4.2 Symmetry-based Image Registration

The key to successful SfM computation is correctly establishing image correspondences. For images of facades with dominant symmetries, correspondence extraction is difficult as repeating structures create multiple *locally consistent* matches, many of which are wrong. Hence, we establish globally consistent correspondences by explicitly using the extracted repetition information. Intuitively, our goal is to position each rectified image on a regularity grid with spacing as extracted in Section 4.1. Using the current symmetry estimates for each image, positioning the images amounts to assigning discrete index positions on the repetition lattice. In other words, the transformation between the images amounts to calculating shifts in rows and columns required to align the corresponding grids in the images. In order to consistently assign these shifts, we collectively analyze the images as described next.

For each image pair (I_i, I_j), a candidate alignment can be encoded as the number of rows and columns the corresponding grids should be shifted over each other. Using the detected grids, all possible candidate alignments between I_i and I_j can be listed as the different shifts in rows and columns. Our goal is to rank such candidates to find the most likely alignment for the image pair. We use SIFT feature matches detected in the original images to perform this evaluation. The detected SIFT features can be grouped into two categories: (i) features that are found outside the repetition regions are most likely to help disambiguate the correct alignment between an image pair; and (ii) features that are detected inside the repetition regions are likely to result in ambiguous matches. In practice, we observe that grid regions also contain discriminating features due

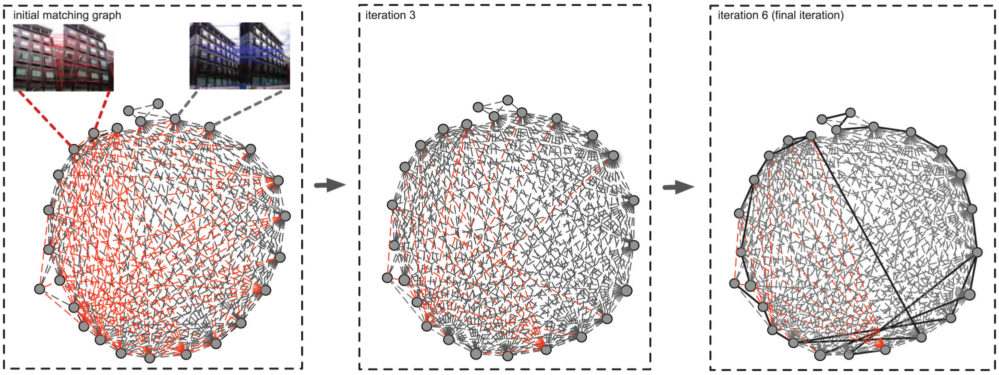

Fig. 6. Using an iterative linear optimization, we compute the global image alignments. Starting from all the candidate pairwise matches, at each iteration we assign costs to the candidate alignments and replace the alignments with high costs by composing alignments along the shortest paths in the graph. Effectively, the quality of the alignments is improved (the wrong alignments at each iteration are in red) and the minimum spanning tree of the final graph (shown as solid edges) provides the image alignments used in the successive steps of the pipeline.

to small random variations such as the shape of curtains, window customization, etc. Therefore, instead of discarding all the features detected inside the repetition regions we only discard those features that are most likely to cause ambiguous matches. Specifically, each candidate alignment between an image pair overlaps certain grid regions according to the encoded shift in rows and columns. We assume that a correct alignment will map the grid regions that are most similar in both images. Further, the feature matches obtained from the remaining regions would agree with the candidate alignment. Therefore, for each candidate alignment, we discard the features detected only inside the overlapped grid cells and match the remaining features. Each pair of feature matches produces an estimate of the scale and translation that maps the corresponding rectified images on which the grids lie. We convert each of these estimates to shifts in rows and columns between the two grids by computing the grid cells they map. (Two grid cells are considered to be mapped to each other if the distance between the cell centers after mapping is below a certain threshold, 2% of the image width in our experiments.) The feature matches, which suggest the current row/column shift being evaluated, provide support for this candidate (see Figure 5). For each image pair (I_i, I_j), we pick the candidate alignment that receives the highest support and assign a weight w_{ij} equal to the fraction of supporting feature matches. After all pairwise candidate alignments are detected, we normalize the support weights of the alignments by dividing by the maximum number of support matches.

We observe that exploring the grid information during feature matching improves the quality of the detected candidate alignments. However, considering only local pairwise relations is unlikely to resolve all the ambiguities. Instead, we analyze collections of pairwise relations as described next.

Image matching graph. We encode the detected pairwise matches as a *matching graph* $G := (V, E)$ where the nodes $n_i \in V$ represent images I_i, while the edges $e_{ij} \in E$ represent the alignment picked between image pairs (I_i, I_j). Each edge is weighted by the corresponding support w_{ij} measured as the fraction of supporting feature matches. We note that such edges in this graph are constructed using information only from image pairs, and hence can contain spurious matches due to ambiguity arising from repetitions. In order to detect such spurious edges, we look for consistency among edges in cycles in this matching graph to assess the reliability of the image alignments (see also Heath et al. [2010]). Specifically, since accumulated alignments along the edges in any cycle in the graph represent a mapping from an image to itself, the corresponding cumulative transform should be the identity, that is, accumulated corresponding grid shifts should result in zero row and column shift. Thus, any cycle where the accumulated shifts do not cancel out indicates the presence of at least one incorrect alignment edge in the cycle. We call such cycles *inconsistent*. Our task is to identify such spurious edges in inconsistent cycles and remove them from G, while still retaining the consistent alignments.

Optimization setup. Based on the preceding observation, we now select a consistent set of alignment edges while discarding the wrong alignments. Effectively, we identify the wrong alignments based on the corresponding supporting weights and the inconsistencies involved. We introduce a binary penalty cost $\chi_e \in \{0, 1\}$ for each edge $e \in E$, where a penalty cost of 1 denotes a wrong alignment and a penalty cost of 0 indicates a correct alignment. Our goal is to extract a globally consistent penalty cost assignment for all the edges in G via a joint formulation. We extract such a set of consistent assignment of costs $\{\chi_e\}$ for edges $e \in E$ as

$$\min_{\{\chi_e\}} \sum_{e \in E} w_e \chi_e$$
$$\text{subject to} \sum_{e \in L_i} \chi_e \geq 1, \quad \forall L_i \in L, \tag{1}$$

where L denotes all the detected inconsistent cycles in the graph G. In our implementation, we only look at 3-cycles in the matching graph G to save computation (see also Nguyen et al. [2011]).

In order to make the aforesaid integer problem convex, we relax the constraints $\chi_e \in \{0, 1\}$ to be $\chi_e \in [0, 1]$. We solve the resultant problem using CVX, a package for specifying and solving convex programs [Grant and Boyd 2008; CVX Research 2012].

In order to map the solution of the optimization to a binary labeling problem, we propose an iterative approach. Intuitively, we first resolve 3-cycles in the graph G and then use the extracted consistent edges to improve estimates of the other alignments. In order to balance between global consistency and local feature-level image-pair matching, we discourage rejection of high-confidence edges, that is, edges with high weights. Hence, we normalize the computed edge penalties as $\chi'_e \leftarrow \chi_e / w_e$ for all the edges. (We update the range of χ_e to be [0.1, 1] to avoid getting continuous penalties of 0.) Next, we use the current edge costs χ'_e to improve image alignments. Specifically, for any edge $e_{ij} \in E$ we compute the shortest path between image nodes n_i, n_j using the edge costs $\{\chi_e'\}$ in G. If the cost of such a shortest path is less than the cost of the original edge, we replace the alignment denoted by e_{ij} by *composing* the alignments along the shortest path, thus potentially improving the alignment for images I_i, I_j (see Figure 7). We also update the weight w_{ij} to be the minimum of the weights along the shortest path. Note that the alignments replaced in this manner implicitly represent longer paths and hence the 3-cycles considered in the successive iterations actually end up as longer cycles in the original graph (see also Nguyen et al. [2011]). After performing the necessary alignment replacements, we resolve a new global optimization using Eq. (1) and continue the process.

This iterative algorithm converges when no more alignment gets replaced (8–10 iterations in all our examples). After convergence, we select the minimum spanning tree of G based on the final edge costs $\{\chi_e'\}$ to obtain the final image alignments (see Figure 6). We use these alignments to filter the pairwise feature matches and preserve only the matches that support the corresponding alignments.

Nongrid alignment. In case of images where no grid is detected or a candidate grid matching alignment with sufficient support (a minimum of 20 supporting feature matches in our tests) is not found, we perform traditional SIFT feature matching and encode the candidate alignment as a transformation matrix. In order to evaluate the consistency of a cycle involving nongrid alignments, we compute the rotation associated with each of the candidate alignments and compute the composite rotation R_L along the cycle (see Zach et al. [2010]). In a consistent cycle, R_L should be equivalent to the identity transformation. However, in a noisy setting this equivalence holds only approximately. We compute the rotation angle α_L of R_L using a quaternion representation and mark the cycle as inconsistent if $|\alpha_L|$ is greater than a threshold angle (10° in our tests). In our datasets, such nongrid alignments were invoked for the images that view the corners of a building with multiple facades.

4.3 Symmetry-Based Structure-from-Motion

In Section 4.2, we established globally consistent feature correspondences across the input images. In addition, we computed the center point of each extracted grid element and form correspondences across these points over all the images. We use these correspondences to estimate the extrinsic camera parameters, while using the EXIF tags to obtain the focal length of the cameras. Finally, using these initial camera parameters and the grid point correspondences across the images, we obtain rough 3D grid points and corresponding grid generators. Note that at this stage, both the camera parameters and the estimated transform generators are only approximate. We refine these parameters by a nonlinear bundle adjustment algorithm, as described next.

We first organize the feature matches and the grid correspondences between the images into *tracks* where each track represents a connected set of matching points across the images. Given such a set of feature tracks and the estimated camera parameters, our goal is to recover the 3D position corresponding to each track. We refine the camera positions and orientations to minimize the reprojection errors, that is, the distance between the projection of a track point and its corresponding image matches. More importantly, we use the estimated symmetry relations between the grid points as a regularizer. Thus, instead of independently computing each 3D grid point, we simultaneously look for the 3D grid parameters, the position of a reference point, and the generators, which minimizes the sum of the reprojection errors of all the grid tracks.

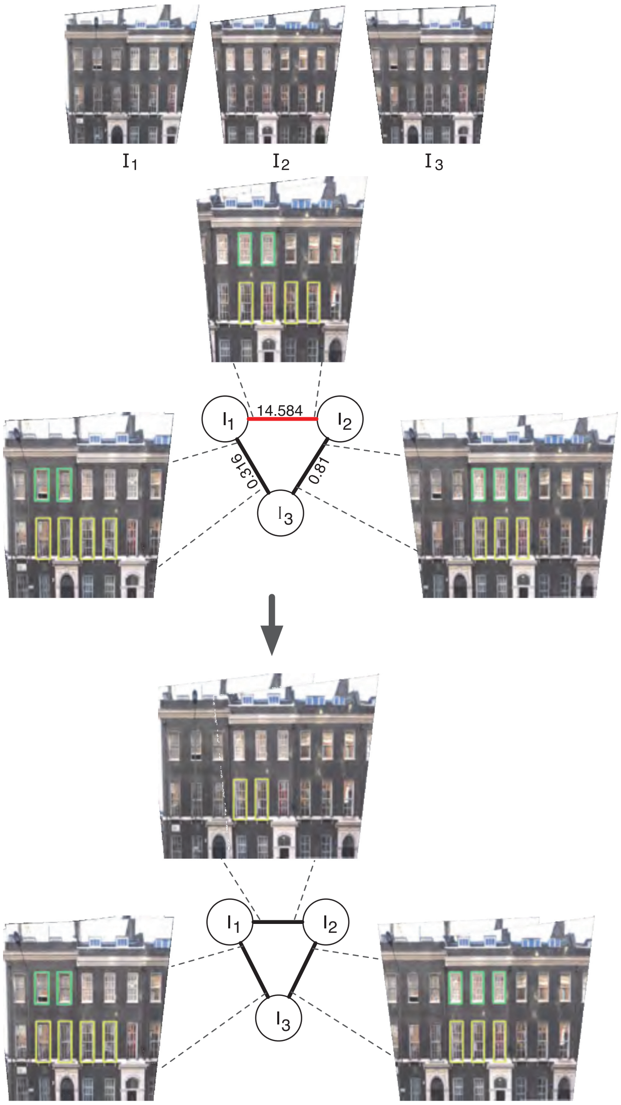

Fig. 7. For the images I_1, I_2, I_3 (in top), the wrong candidate alignment between (I_1, I_2) (in red) is replaced by the correct accumulated alignment along the shortest path $I_1 \to I_3 \to I_2$ (the edge cost $\{\chi_e'\}$ for each alignment is given) during the iterative grid optimization. For each alignment, the overlapped images are shown together with the mapped grid regions (in yellow and green).

Assume we have a set of m cameras parameterized by A_k. Let $P(A_k, \mathbf{p})$ denote the projection function mapping a 3D point $\mathbf{p}$ to its 2D projection $\mathbf{q}$ in the k-th camera with parameters A_k. Furthermore, assume a grid with r rows and c columns is represented by the reference point $\mathbf{o}$ and the grid generators $(\mathbf{t}_h, \mathbf{t}_v)$. Minimizing the reprojection error of the grid tracks corresponding to this 3D grid is equivalent to minimizing the following energy:

$$E_{grid} := \sum_{k=1}^{m} \sum_{i=1}^{r} \sum_{j=1}^{c} \lambda_{ij}^{k} \|\mathbf{q}_{ij}^{k} - P(A_k, \mathbf{x}_{ij})\|, \qquad (2)$$

where the 3D grid point at the i-th row and j-th column of the grid is represented as $\mathbf{x}_{ij} = \mathbf{o} + (i-1)\mathbf{t}_h + (j-1)\mathbf{t}_v$. $\mathbf{q}_{ij}^{k}$ denotes the projection of this point in the k-th camera. λ_{ij}^k is an indicator variable equal to 1 if this grid point is visible in the camera and 0 otherwise.

Moreover, assume we have n nongrid feature tracks parameterized by the 3D points $\mathbf{b}_i$. Minimizing the sum of the reprojection errors for these tracks is equivalent to minimizing the following energy:

$$E_{other} := \sum_{k=1}^{m}\sum_{i=1}^{n} \beta_i^k \|\mathbf{q}_i^k - P(A_k, \mathbf{b}_i)\|, \tag{3}$$

where β_i^k is an indicator variable with $\beta_i^k = 1$ if point $\mathbf{b}_i$ is visible in the k^{th} camera and 0 otherwise.

Finally, we combine the energy terms for the 3D grid and the nongrid tracks and minimize the resulting objective function using the Levenberg-Marquardt method (see Lourakis et al. [2009]):

$$E := E_{grid} + E_{other}. \tag{4}$$

At the end of this enhanced bundle adjustment step, we obtain the refined camera parameters, a sparse 3D representation of the scene, and the refined grid parameters.

4.4 Vanishing Line Refinement

The results obtained at this stage retain any bias introduced in the initial vanishing line detection used for rectification. To reduce this bias, we update the rectification of the input images using the computed 3D geometry. Specifically, we fit planes to the reconstructed 3D point cloud where each plane corresponds to a facade of the building. Then we find the facade plane on which each detected 3D grid lies and update the position of each 3D point representing a grid cell by projecting to its facade plane. We back-project these 3D points to the input images to refine the grid correspondences between the images. Please note that, in case the repeating facade elements are found at a larger depth offset than the facade plane, small shifts might be observed in the location of the grid correspondences in the nonfrontal viewpoints. However, in our experiments we did not observe this to create serious artifacts. Experimentally we found that any minor errors that might have occurred during the grouping of the image grids (due to small variations in grid transformations which have not been detected in image space) are fixed. For each 3D point $\mathbf{x}$ representing a grid cell in a 3D grid, we compute the average reprojection error as

$$P_{error} := \sum_{k=1}^{m} \lambda^k \|\mathbf{q}^k - P(A_k, \mathbf{x})\| / \sum_{k=1}^{m} \lambda^k, \tag{5}$$

where λ^k is an indicator variable equal to 1 if this grid point has been detected in the k-th image, and $\mathbf{q}^k$ denotes this detected 2D grid point if λ^k is 1. We exclude the grid cells for which P_{error} is above a certain threshold (5% of image width in our experiments) and update the number of rows and columns of a grid accordingly. With these refined correspondences, we rerun the constrained bundle adjustment step to obtain the final camera parameters and the 3D grids. In our tests, we found a single iteration of refinement to be sufficient (see Figure 8).

4.5 Extension to Multiple Grids

We now describe how we adapt our pipeline to handle multiple grids, which can occur as: (i) multiple arrangements of the same base element, or (ii) grids of different user-indicated base elements.

Given any image pair (I_i, I_j), let us assume multiple grids have been detected in these images. Any grid pair (g_i, g_j), where $g_i \in I_i$ and $g_j \in I_j$, that shares the same base element is a potential projection of the same 3D grid. Therefore, during the image feature matching step, we list all the potential matching grid pairs across the images. For each such grid pair (g_i, g_j), we perform the symmetry-guided feature matching step as explained in Section 4.2. Specifically, we evaluate all the candidate alignments corresponding to different shifts in rows and columns between the grids g_i and g_j. Once the candidate alignments have been evaluated for all the grid pairs, we pick the alignment with the highest support. Using the feature matches that support this selected alignment, we detect the remaining matching grids in the images (I_i, I_j) that are the projection of the same 3D grid and find the corresponding row/column shifts between them. We add the selected alignment to the matching graph G as edge e_{ij} encoding the grid shifts between all the matching grids between the corresponding images.

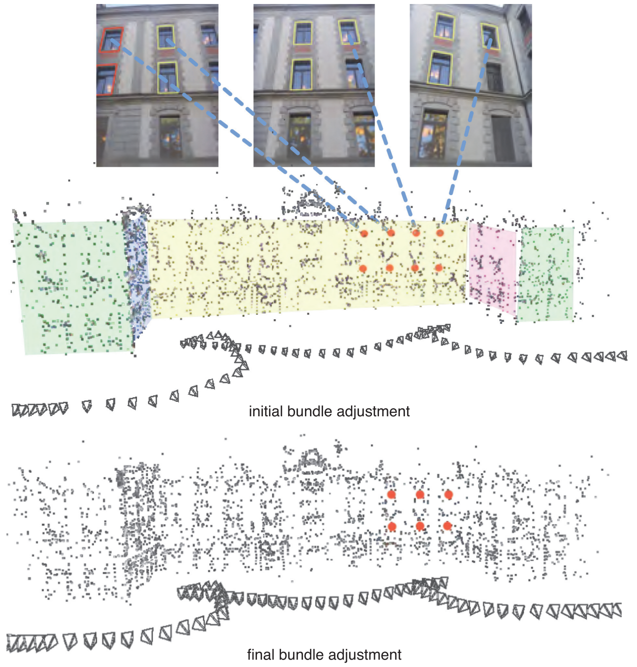

Fig. 8. Once we obtain the initial 3D scene representation, we refine the rectification of the input images and update the grid correspondences by fitting 3D planes to the reconstructed geometry (each fitted plane is shown in a different color). We compute the average reprojection error of each 3D grid point and discard the grid cells with high errors. In the initial image matching step, the windows in red have been grouped together with the windows in yellow (resulting in a 2-by-4 grid) but have been discarded due to high reprojection error (resulting in a 2-by-3 grid). We perform the constrained bundle adjustment with the updated correspondences to compute the refined 3D scene representation.

Later, in the iterative optimization setup, for each cycle in the image matching graph, we consider the accumulated alignments between all the common grids in the images participating in the cycle. More specifically, for a cycle between the images I_i, I_j, and I_k, grids $g_i \in I_i$, $g_j \in I_j$, and $g_k \in I_k$ are common grids if the edge e_{ij} encodes a grid shift between g_i and g_j, e_{jk} encodes a grid shift between g_j and g_k, and e_{ki} encodes a grid shift between g_k and g_i. If the accumulated grid shifts between any such common grid do not cancel out, the cycle is marked as *inconsistent*. Similarly, if an alignment is to be swapped with an accumulated alignment along the shortest paths in G, all the grid shifts represented by the edge

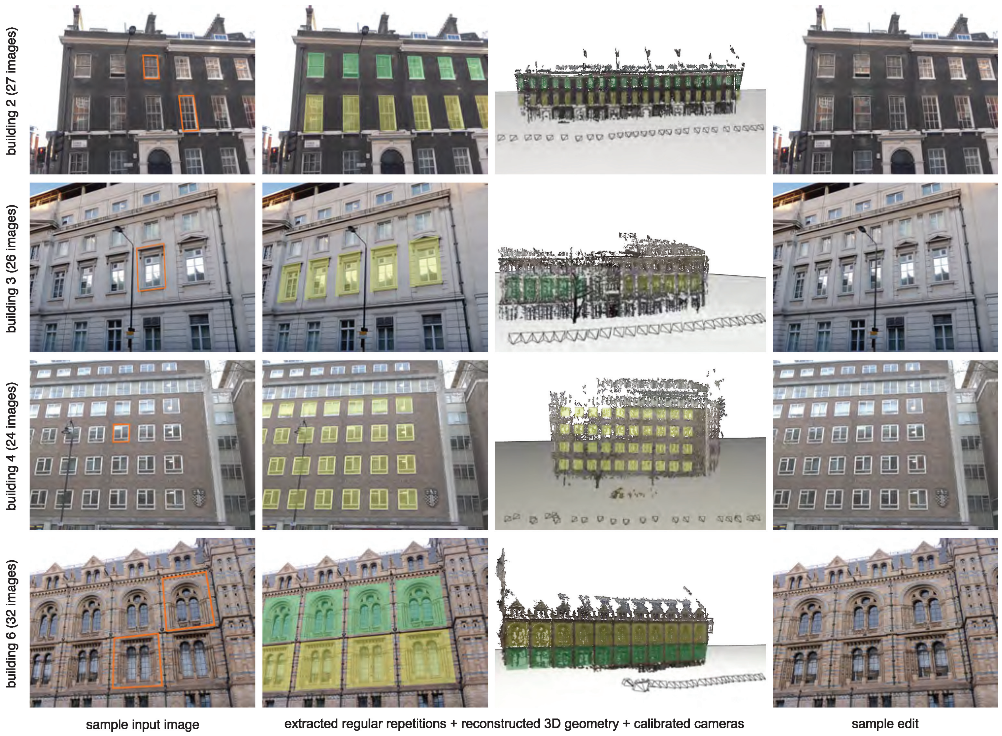

Fig. 9. Starting from a set of input images and a sample repeated element(s) marked on a single image (shown in orange), we simultaneously recover the 3D repetition patterns, calibrate the cameras, and reconstruct the scene geometry. The information is then used for a range of coupled image manipulations (see supplementary for complete results, comparisons, and applications).

between the common grids in the images involved in the path get updated.

At the end of the iterative optimization, the final alignments computed for each image pair (I_i, I_j) encode the correct matching image grids and the shifts between them. Using these alignments, all the image grids matched to each other across the input images are grouped together where each group represents the image projections of the same 3D grid. For each 3D grid, we organize the grid correspondences as *grid tracks*. We update the bundle adjustment objective function to include a grid energy term for each of the g 3D grids and refine the parameters of each 3D grid with this bundle adjustment step:

$$E := \sum_{1}^{g} E_{grid} + E_{other}. \tag{6}$$

Figure 11 shows the results in presence of multiple facades.

5. EVALUATION

We evaluate our framework on several datasets with varying complexity of the underlying symmetries. We now summarize our main findings, while referring to the supplementary material for complete results and comparisons with other methods. Table I lists the performance statistics of our algorithm for each dataset.

In our framework, we explicitly detect repeating elements in the input images and use this information both to extract reliable image correspondences and estimate camera parameters accurately. We compare this approach to a standard SfM pipeline [Snavely et al. 2006] and the inference-based ambiguity detection method of Zach et al. [2010], which also has been used as an initializer in the follow-up work of Cohen et al. [2012]. To illustrate the effect of accurate camera pose estimation for dense reconstruction, we use a state-of-the-art multiview stereo method [Furukawa and Ponce 2007] to

Table I. The Table Shows the Number of Input Images (N_i), the Resolution of the Images in Megapixels (res), and the Total Number of Repeating Elements Detected (N_r) for Each Dataset

	N_i	res	N_r	T_s (mins)	T_o (mins)	Ours	Bundler	Zach et al. 10
Bld. 1	26	6.2	28	10	5	yes	poor	no
Bld. 2	27	7.7	42	15	2	yes	conf.	poor
Bld. 3	26	3.5	10	8	2	yes	yes	yes
Bld. 4	24	7.7	44	6	1.5	yes	yes	yes
Bld. 5	25	6.2	22	20	1.5	yes	conf.	poor
Bld. 6	32	6.2	22	40	1	yes	yes	poor
Bld. 7	51	6.2	101	45	6	yes	mult..	mult.
Bld. 8	72	6.2	36	50	8	yes	no	mult.
Bld. 9	13	6.2	0	4	1	no	no	no
Bld. 4 lowRes	24	0.5	44	1	0.5	yes	no	conf.

We also report how our method, Bundler, and the method of Zach et al. perform: a correct reconstruction is produced (*yes*), the output is poor in quality (*poor*), there is a confusion in the number of repeated elements (*conf.*), or reconstruction contains multiple misaligned components (*mult.*). The computation times for image-based symmetry detection (T_s) and a single iteration of symmetry-based SfM (T_o) are given in minutes measured on a 2.8 GHz 4-core machine.

produce dense reconstructions of the input scenes using the camera parameters estimated by each of the methods. We use the EXIF tags of the images to estimate the focal length in all three cases. In Table I, we report how these three methods perform on each dataset marking the output based on if: (i) a correct reconstruction is obtained, (ii) output is poor in quality, (iii) wrong number of repeated elements is reconstructed, or (iv) multiple submodels corresponding to different subsets of the input images are reconstructed.

Our algorithm accurately extracts the camera parameters in most of the examples leading to accurate dense reconstructions, while *Bundler* [Snavely et al. 2006] or the approach of Zach et al. [2010] fail or often produce sparser reconstructions (see Figure 9 and supplementary materials). Explicit encoding of the extracted symmetry information enables our method to effectively evaluate the reliability of the alignments between the input image pairs. In the symmetry-guided feature matching step, we first list all the candidate alignments between an image pair, and then discard the potential ambiguous matches. We observe that the remaining sparse set of feature matches arising due to nonrepeating regions and the random variations in the nondiscarded repeating regions (such as ornaments, weathering, etc.) often provide sufficient support for the correct alignments.

Moreover, during the global optimization performed on the matching graph, we iteratively improve the quality of the alignments based on the supporting feature matching weights and by discarding the discovered inconsistencies. Figure 10 illustrates the effectiveness of the grid constraints during the iterative graph optimization step. As a result, we achieve robustness even in the presence of large number of wrong pairwise alignments (see Figure 6) which otherwise inhibits producing a reasonable output or leads to confusion in the number of repetition elements.

Figure 11 shows performance of our method on multiple facades of buildings. Please note that we do not make any assumption about orthogonality across facade faces.

Comparison. In Figure 12, we compare our method to the recent approach of Jiang et al. [2012] that formulates the image matching problem as finding the spanning tree of the image matching graph minimizing a global energy function. They propose a greedy search algorithm that resolves an important portion of the image matching ambiguities. They do not explicitly model any particular form of symmetry or repetition. Instead, by injecting symmetry priors into every step of the reconstruction pipeline, our method effectively resolves the remaining ambiguities and significantly improves the quality of the reconstructions. Further, in contrast to all other methods, our approach produces the repetition patterns directly as part of the output.

Symmetry as regularizer. Figure 13 illustrates the effect of using additional symmetry constraints in bundle adjustment on the quality of the final reconstructions. Given a set of feature tracks, explicitly enforcing the symmetry relations across the grid correspondences acts as a regularizer and significantly reduces drift, especially in long image sequences. Further, our algorithm successfully recovers the correct orientation of the individual facade planes of a building *without* any additional assumption on the orientation relations like orthogonality.

Robustness to low-resolution inputs. Our symmetry-aware correspondence search makes the approach robust to degrading image resolution. To illustrate this, we tested our framework on an image set at two resolutions (see Figure 14). Both Bundler and the method of Zach et al. [2010] performed poorly in the low-resolution setting because a significant amount of features are only seen in the high-resolution images due to small random variations in the facade elements. However, our method extracts the correct relations among the input cameras using the sparse feature set by exploring the initially detected repetitions in the images.

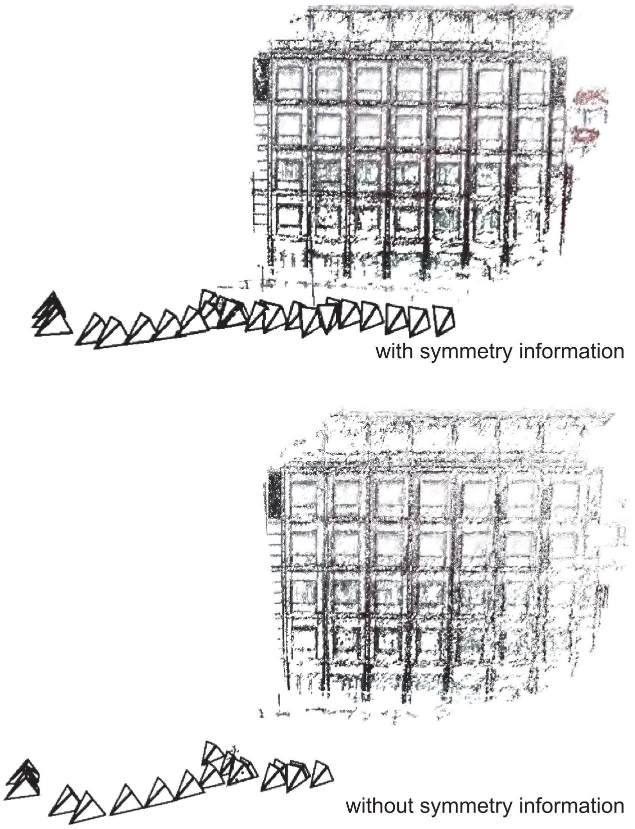

Fig. 10. (Top) Explicit encoding of the symmetry information effectively resolves the image alignment ambiguities during 3D reconstruction. (Bottom) If the symmetry information is discarded during the iterative graph optimization and the consistency between the image alignments is evaluated based on the corresponding rotations only, some ambiguities remain unresolved and only a subset (18 of 26) of the input image set is calibrated.

Limitations. Although we handle a range of diverse datasets, the approach still suffers from the following limitations: We assume facade elements to be repeated along 1- or 2-parameter regular grids and do not handle rotational symmetries as found on domes, churches, etc. In certain cases repeated elements have nonuniform gaps with sufficiently small variations that cannot be recovered neither in image space nor the 3D reconstruction step (see Figure 15). Further, repetitions might be arranged in skewed distributions. As a result, image-based regular structure detection fails and our method effectively degenerates to standard SfM approaches.

Even though the symmetry-guided correspondence search and the iterative global optimization improves the quality of the image alignments significantly, we do require a sufficient amount of discriminating features to bootstrap the process (see Figure 16). Although we do not require the repetition grids to be visible in full from any single image, we do expect a reasonable overlap between the images so that we get a connected solution graph. In the absence of sufficient overlap, the solution graph can have multiple components.

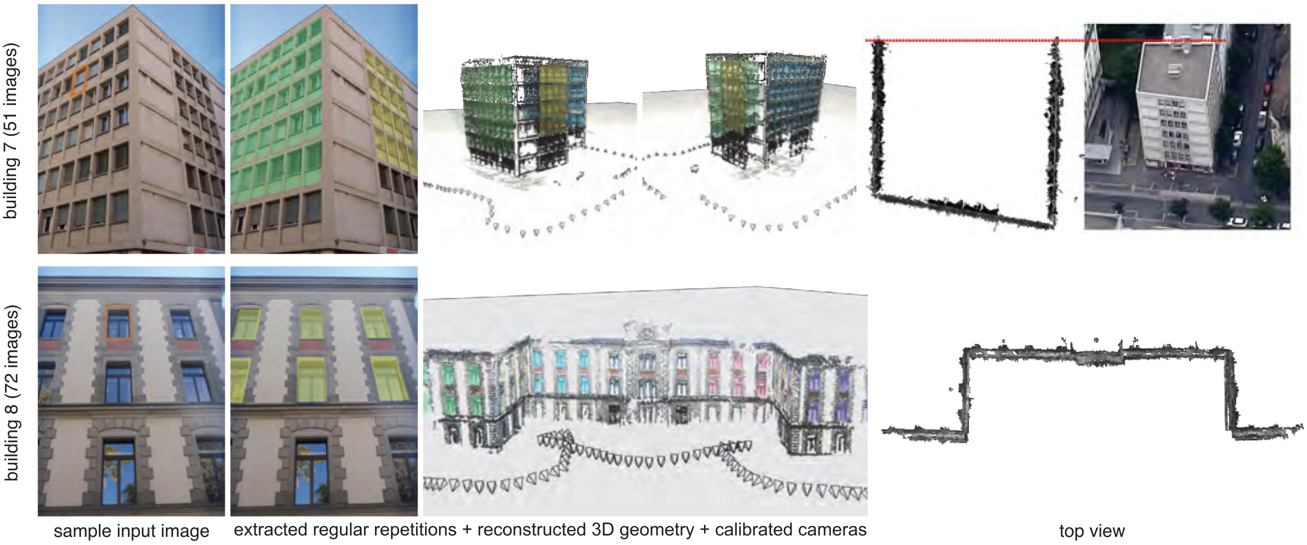
building 7 (51 images)
building 8 (72 images)
sample input image
extracted regular repetitions + reconstructed 3D geometry + calibrated cameras
top view

Fig. 11. Our approach can successfully handle buildings with multiple facades and recover the 3D repetition pattern on each facade. Note that 3D geometry extracted using grid constraints preserves the correct orientation of the individual facades both for orthogonal (*building 8*) and nonorthogonal (*building 7*) relations. We provide a satellite imagery of *building 7* for reference.

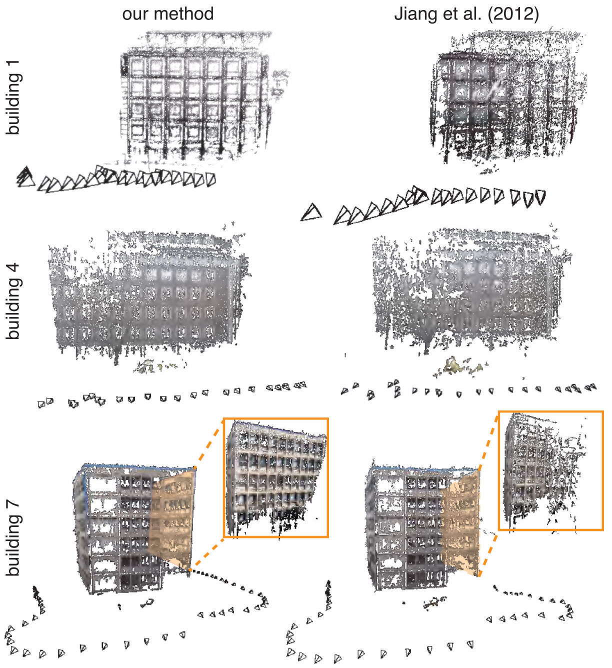

Fig. 12. Comparison with the method of Jiang et al. [2012]. For the *building 1* dataset, their method registers 21 out of 26 images. On these three examples, our method, which implicitly considers facade-level symmetry, distributed the error evenly across the facades. This effect is particularly visible on *building 7* dataset, where our method produces significantly higher-quality output especially for the right facade of the building highlighted in orange. By directly considering symmetry information, our method robustly prevents drift effects on facades with many repeated elements.

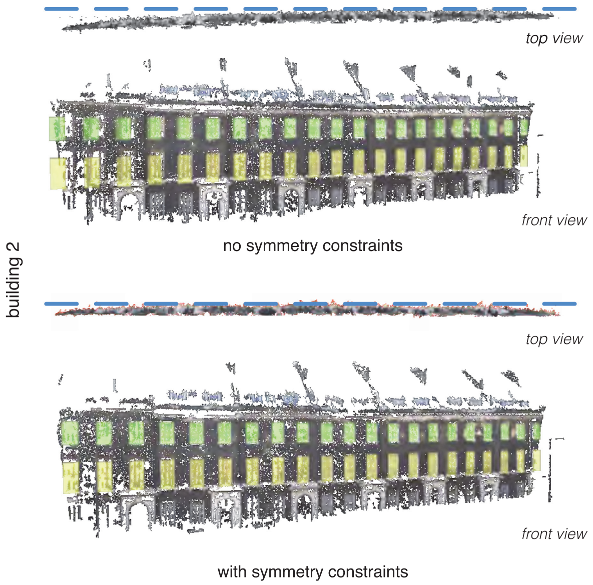

Fig. 13. Global constraints lead to improved SfM reconstructions: 3D geometry extracted using correspondences recovered from the constructed solution graph without and with nonlocal symmetry prior. Note that the unconstrained solution produces a distorted facade (blue dotted line shown for reference) with the repeated elements drifting from the correct solution.

While we did not encounter such a problem in our examples, in the future we want to better characterize the requirements on the input images. Finally, we focus on facades with dominant facade planes. If facade elements show significant depth variations, image-based repetition detection cannot be performed reliably [Jiang et al. 2011]. Focusing on planar facades, we obtain limited depth information in

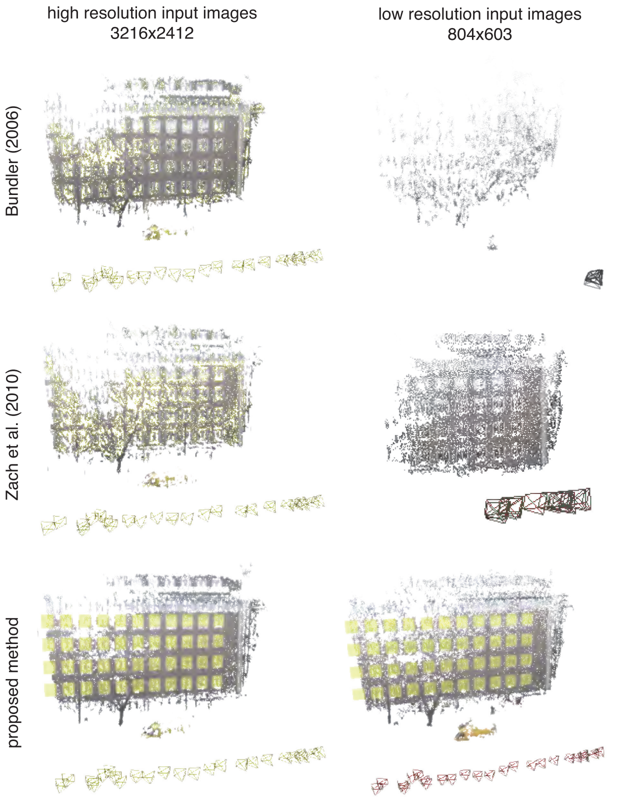

Fig. 14. Our method to simultaneously extract scene repetitions and 3D geometry is robust even with low-resolution input images. In contrast, popular methods like Bundler [Snavely et al. 2006] or state-of-the-art alternatives like Zach et al. [2010] degrade quickly. We also show the recovered camera positions for comparison, with ground-truth camera path being near linear.

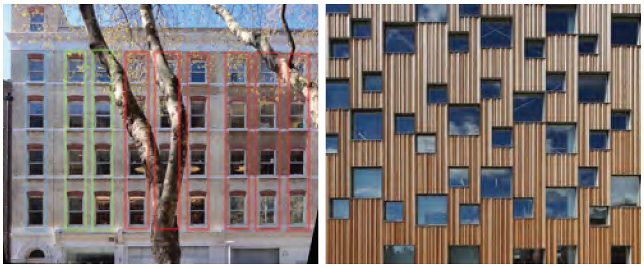

Fig. 15. In absence of detected regular 1- or 2-parameter repetitions (as in the right image), our algorithm degenerates to SfM methods. The uneven spacing among the marked elements in the left image prevents detection of the *near regularity*.

our 3D reconstructions, especially around sharp features, statues, etc., which limits the scope of subsequent editing possibilities.

6. APPLICATIONS

Once we register the set of input images and extract the symmetries of the scene, our system allows the user to perform various image editing applications, while automatically propagating the changes to all the images, as described next. Besides demonstrating various use scenarios, these applications are important to highlight the robustness and accuracy of the reconstruction algorithm as the applications heavily rely on precise symmetry boundaries.

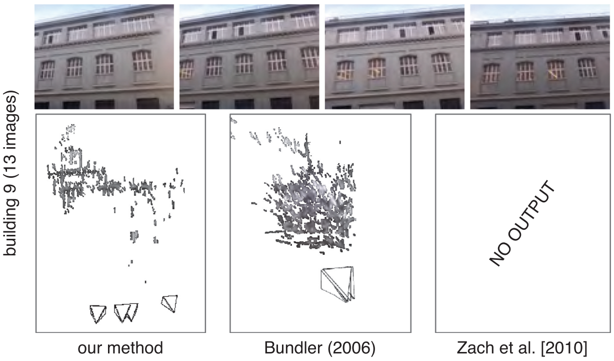

Fig. 16. For the dataset where sample input images are shown (top), due to lack of sufficient discriminating feature matches our method fails to resolve the ambiguities. The dense reconstructions computed with the camera parameters obtained from our method and Bundler are shown (bottom). The method of Zach et al. [2010] does not produce any camera parameters.

Occlusion Removal. Often street-level images of building facades contain many foreground objects such as street lights, trees, and cars, which partially occlude the background facade plane. Redundancy in the form of multiple images and repeated elements in the input set allows us to synthesize seamless textures to remove such occlusions.

Given a set of input images $I_1, \ldots, I_n$, we require the user to provide rough strokes on a single reference image I_i to denote the occluding object that is to be removed. We use GrabCut [Rother et al. 2004] to extract an accurate mask B_i for the occlusion area. (Note that if the occluder object has delicate structures, like tree leaves, we require the user to provide more refined strokes.) Typically, each pixel in this bounded region occludes a point that lies on the main facade plane. Our goal is to synthesize the texture for this occluded area of the facade plane, which we approximate by using the 3D grid information extracted in Section 4.

We back-project each pixel $p \in B_i$ in the occlusion region to the facade plane to obtain the 3D position P of the corresponding occluded point. We then project P to the other images to find a set of candidate pixel colors that can be used to replace p in the reference image. Once a set of candidate colors has been computed for each pixel, the desired texture is synthesized by estimating a label $L(p) \in [1, \ldots, n]$ for each pixel p denoting which image should be used as the source color for p. We formulate this texture synthesis problem as a Markov Random Field (MRF) optimization by minimizing an energy function consisting of data and smoothness terms. The data term $E_{data}(p, L(p))$ defines the cost of assigning the label $L(p)$ to pixel p and is defined as $E_{data}(p, L(p)) = \mid I_{L(p)}(p) - \rho \mid$, where $I_{L(p)}$ denotes the color of pixel p in the labeled image and ρ is the median of the candidate colors for pixel p. To reduce the seams in the synthesized texture, we define a smoothness term to asses the labelings of neighboring pixels by evaluating the color and the gradient differences similar to Sinha et al. [2008]. We use the graph cut algorithm [Boykov et al. 2001] to optimize the defined energy function and choose the source images for each pixel in the synthesized texture. Finally, we use Poisson blending [Pérez et al. 2003] to further reduce the seams across the occlusion boundaries in the original image (Figure 17).

Importantly, once the user-marked occluding area B_i is removed from the reference image, we can propagate the edit and remove the

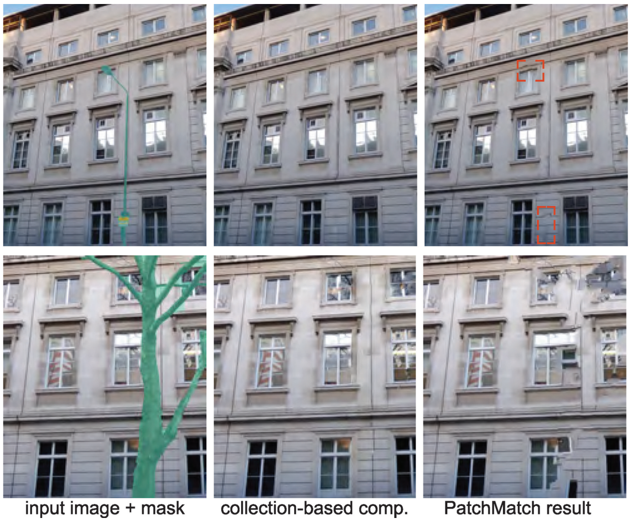

Fig. 17. Occlusion removal using facade information propagated from other images. In contrast, single-image-based state-of-the-art PatchMatch [Barnes et al. 2009] can lead to artifacts as highlighted in red (see supplementary for additional examples).

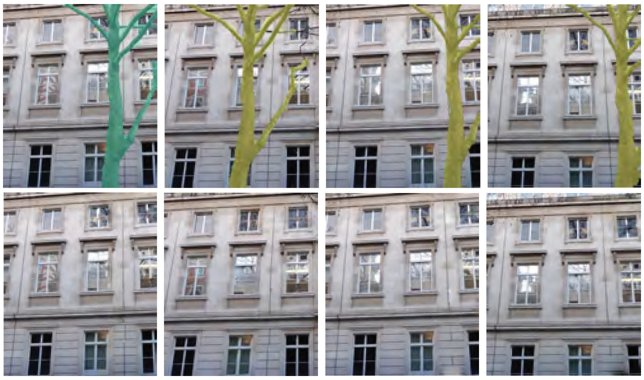

Fig. 18. Occlusion mask marked in one image (top-left) is propagated to the other images using the reconstructed 3D geometry (top row). Subsequently, we use the reconstructed 3D facade geometry to complete the occluded regions. See supplementary material for additional examples.

occluder also from the other images. To enable this operation, we need depth estimates for the occlusion object in the reference view to determine its projection in the other images. As the occluding object lies in front of the main facade plane, we construct a depth volume bounded by the reference camera position and the depth of the facade plane and use an MRF energy minimization approach to optimize for the depth of each pixel in the occluded area. We formulate the data term of the MRF energy based on NCC matching costs of image patches (see Campbell et al. [2008]) and define the smoothness energy as the weighted sum of the depth deviation between a pixel and its 4-connected neighbors [Szeliski et al. 2008]. Similar to Sasaki et al. [2006], we define the data term in a multiresolution framework to increase the accuracy of the depth estimates. Once the depth estimates for each pixel in the reference occluding region are computed, this region is projected to the other images and the texture is synthesized in these projected areas as described before (see Figure 18 and supplementary material).

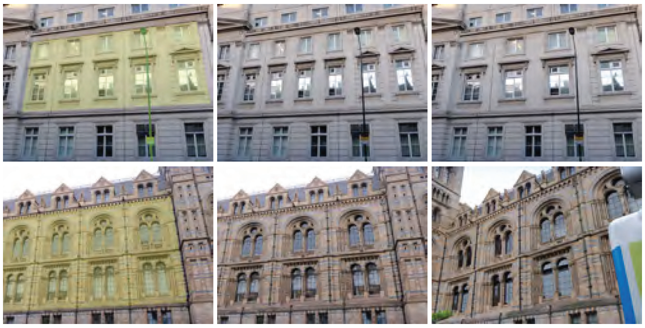

Fig. 19. (Top) Extracted facade grid patterns are changed and then composited with the foreground (e.g., lamp post); (bottom) repetition pattern manipulated and the changes propagated to the other images (see supplementary for additional examples).

Grid Editing. Our framework extracts 2D/3D symmetry information of the input scene in the form of planar repetition grids. This information allows the user to directly manipulate the grids such as changing the repetition count in the grid as previously shown by Wu et al. [2010b] or editing the appearance of the grid elements on a single image. The system automatically propagates the changes to the other images as the relation between the 2D image grids and the global 3D grid is already computed. Technically, we first extract accurate boundaries of the repeating elements by snapping their initial contours to the image edges and optimizing for the common contour by a line fitting approach similar to Ceylan et al. [2012]. In contrast to their method, we only optimize for the contour lines as we have already extracted the refined 3D grid generators. Once the contour of each grid element is computed, we change the number of repetitions in the 3D grid by keeping the boundaries of the grid fixed and appropriately scaling the grid elements. This amounts to editing only the relevant parts of each image where the 3D grid projects (see Figure 19). We synthesize the texture for the new grid elements by scaling the original elements. To minimize seams and lighting variations, for each new grid element, we use the texture from the spatially closest original grid element. In case of occlusion, we require the user to roughly annotate the occluding object, while we perform necessary image completion as described before. The occluding object is encoded as a separate layer and later composed with the edited images (see Figure 19).

Limitations. The proposed consistent image editing applications are only as powerful as the detected symmetries. Also, lack of sufficient depth information can lead to artifacts near 3D elements on the original images (e.g., window ledges, etc.). Thin structures (e.g., tree branches, leaves) can be difficult to annotate in the images and also challenging to propagate across images since corresponding 3D points are few and sparse. Finally, grid editing can lead to semantic ambiguities across the grid boundaries, for example, the drainpipes in the museum example (see Figure 19).

7. CONCLUSION

We presented a structure-from-motion framework that detects and conforms to structural regularities, while simultaneously recovering 3D geometry starting from a set of facade images. A novel graph-based global analysis yields a globally consistent 3D geometry reconstruction with explicit encoding of the facade regularities. These regularities can then be used for a range of novel image manipulations, while maintaining consistency across the images.

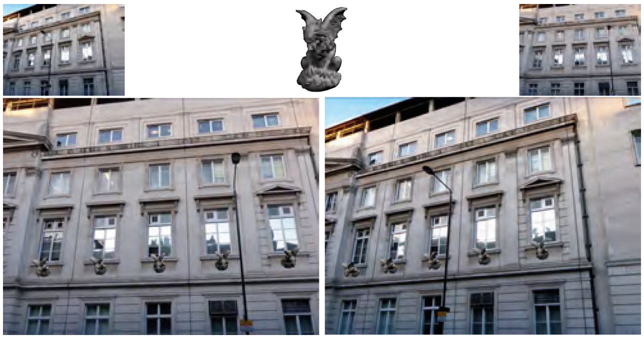

Fig. 20. Future work. The extracted scene geometry along with symmetry information can be used towards new editing possibilities. In this example, we can add a synthetic 3D gargoyle model spaced according to the extracted repetitions and keep the changes consistent across the images. Insets show the original images and the inserted statue. Advanced appearance matching can potentially be used for better color blending (e.g., [Zheng et al. 2012]).

Our evaluation shows the robustness and accuracy of the framework on a range of challenging facade image sets.

With the growing demand for simple, fast, and accurate acquisition methods for urban facades in digital cities, we expect to see increasing research efforts in this direction. A natural extension of our approach will be to incorporate rotational symmetries and surfaces of revolution as found on arches, domes, etc. Although in this work we focused on facades with dominant facade planes, in the future, we plan to investigate how to handle depth variations. One approach would be to extend the work of Wu et al. [2011] that extracts the depths of the repeating facade elements in a single view to a multiple-view setting. However, we believe that handling small depth changes due to the presence of architectural features like ornaments, window ledges, etc., requires further investigation.

Moreover, extending this approach to handle building colonies where elements are also repeated across different buildings, or repeated window elements are in different open/close configurations [AlHalawani et al. 2013] are interesting research directions. Finally, the coupled symmetry information and 3D reconstruction suggests novel editing applications by combining information across the images and the sparse 3D geometries, which can lead to interactive and dynamic interfaces when integrated with existing navigation systems such as Google Streetview. We present a first result in Figure 20.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their valuable comments. We are grateful to Minh Dang and Sawsan AlHalawani for proof-reading the article and Charlotte E. Rakhit for the video voice over.

REFERENCES

S. Alhalawani, Y.-L. Yang, H. Liu, and N. J. Mitra. 2013. Interactive facades: Analysis and synthesis of semi-regular facades. *Comput. Graph. Forum* 32, 2pt3, 215–224.

G. Baatz, K. Koser, D. Chen, R. Grzeszczuzuk, and M. Pollefeys. 2010. Handling urban location recognition as a 2D homothetic problem. In *Proceedings of the European Conference on Computer Vision (ECCV'10)*. 266–279.

C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* 28, 3, 24:1–24:11.

A. Bartoli and P. Sturm. 2003. Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene. *Int. J. Comput. Vis.* 52, 45–64.

Y. Boykov, O. Veksler, and R. Zabih. 2001. Fast approximate energy minimization via graph cuts. *IEEE Pattern Anal. Mach. Intell.* 23, 11, 1222–1239.

N. D. Campbell, G. Vogiatzis, C. Hernandez, and R. Cipolla. 2008. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV'08)*. 766–779.

D. Ceylan, N. J. Mitra, H. Li, T. Weise, and M. Pauly. 2012. Factored façade acquisition using symmetric line arrangements. *Comput. Graph. Forum* 31, 1, 671–680.

A. Cohen, C. Zach, S. Sinha, and M. Pollefeys. 2012. Discovering and exploiting 3D symmetries in structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*. 1514–1521.

CVX Research. 2012. CVX: Matlab software for disciplined convex programming, version 2.0 beta. http://cvxr.com/cvx

Y. Furukawa and J. Ponce. 2007. Accurate, dense, and robust multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*. 1–8.

A. Gil, O. Reinoso, O. Mozos, C. Stachnissi, and W. Burgard. 2006. Improving data association in vision-based slam. In *Intelligent Robots and Systems*, 2076–2081.

V. Govindu. 2004. Lie-algebraic averaging for globally consistent motion estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*. 684–691.

V. M. Govind. 2006. Robustness in motion averaging. In *Proceedings of the Asian Conference on Computer Vision (ACCV'06)*. 457–466.

M. Grant and S. Boyd. 2008. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, vol. 371, Springer, 95–110.

K. Heath, N. Gelfand, M. Ovsjanikov, M. Aanjaneya, and L. Guibas. 2010. Image webs: Computing and exploiting connectivity in image collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*. 3432–3439.

N. Jiang, P. Tan, and L.-F. Cheong. 2009. Symmetric architecture modeling with a single image. *ACM Trans. Graph.* 28, 5, 113:1–113:8.

N. Jiang, P. Tan, and L.-F. Cheong. 2011. Multi-view repetitive structure detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'11)*. 535–542.

N. Jiang, P. Tan, and L.-F. Cheong. 2012. Seeing double without confusion: Structure-from-motion in highly ambiguous scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*. 1458–1465.

M. Klopschitz, A. Irschara, G. Reitmayr, and D. Schmalstieg. 2010. Robust incremental structure from motion. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'10)*.

Y. Li, Q. Zheng, A. Sharf, D. Cohen-Or, B. Chen, and N. J. Mitra. 2011. 2D-3D fusion for layer decomposition of urban facades. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'11)*. 882–889.

M. A. Lourakis and A. Argyros. 2009. SBA: A software package for generic sparse bundle adjustment. *ACM Trans. Math. Softw.* 36, 1, 1–30.

D. Martinec and T. Pajdla. 2007. Robust rotation and translation estimation in multiview reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (ICPR'07)*. 1–8.

B. Micusik, H. Wildenauer, and J. Kosecka. 2008. Detection and matching of rectilinear structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*. 1–7.

N. J. Mitra, M. Pauly, M. Wand, and D. Ceylan. 2013. Symmetry in 3D geometry: Extraction and applications. *Comput. Graph. Forum* 32, 6.

P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, I. Van Gool, and W. Purgathofer. 2012. A survey of urban reconstruction. *Comput. Graph. Forum* 32, 6.

L. Nan, A. Sharf, H. Zhang, D. Cohen-Or, and B. Chen. 2010. Smartboxes for interactive urban reconstruction. *ACM Trans. Graph.* 29, 93:1–93:10.

A. Nguyen, M. Ben-Chen, K. Welnicka, Y. Ye, and L. Guibas. 2011. An optimization approach to improving collections of shape maps. *Comput. Graph. Forum* 30, 5.

M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. Guibas. 2008. Discovering structural regularity in 3D geometry. *ACM Trans. Graph.* 27, 3, 43:1–43:11.

P. Perez, M. Gangnet, and A. Blake. 2003. Poisson image editing. *ACM Trans. Graph.* 22, 3, 313–318.

M. Pollefeys, D. Nister, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Cliff, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, and H. Towles. 2008. Detailed real-time urban 3D reconstruction from video. *Int. J. Comput. Vis.* 78, 143–167.

L. Quan and T. Kanade. 2010. *Image-Based Modeling* 1st Ed. Springer.

R. Roberts, S. Sinha, R. Szeliski, and D. Steedly. 2011. Structure from motion for scenes with large duplicate structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. 3137–3144.

C. Rother, V. Kolmogorov, and A. Blake. 2004. "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.* 23, 309–314.

K. Sasaki, S. Kameda, and A. Iwata. 2006. Stereo matching algorithm using a weighted average of costs aggregated by various window sizes. In *Proceedings of the Asian Conference on Computer Vision (ACCV'06)*. 771–780.

S. N. Sinha, D. Steedly, and R. Szeliski. 2010. A multi-stage linear approach to structure from motion. In *Proceedings of the European Conference on Computer Vision Workshop on Reconstruction and Modeling of Large-Scale 3D Virtual Environments*. 267–281.

S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys. 2008. Interactive 3D architectural modeling from unordered photo collections. *ACM Trans. Graph.* 27, 5, 159:1–159:10.

N. Snavely. 2011. Scene reconstruction and visualization from internet photo collections: A survey. *IPSJ Trans. Comput. Vis. Appl.* 3, 44–66.

N. Snavely, S. M. Seitz, and R. Szeliski. 2006. Photo tourism: Exploring photo collections in 3D. *ACM Trans. Graph.* 25, 3, 835–846.

R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. 2008. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Pattern Anal. Mach. Intell.* 30, 6, 1068–1080.

T. Tuytelaars, L. Van Gool, M. Proesmans, and T. Moons. 1998. The cascaded hough transform as an aid in aerial image interpretation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'98)*. 736–739.

G. Wan, N. Snavely, D. Cohen-Or, Q. Zheng, B. Chen, and S. Li. 2012. Sorting unorganized photo sets for urban reconstruction. *Graph. Models* 74, 1, 14–28.

C. Wu, J. Frahm, and M. Pollefeys. 2011. Repetition-based dense single-view reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. 3113–3120.

C. Wu, J.-M. Frahm, and M. Pollefeys. 2010a. Detecting large repetitive structures with salient boundaries. In *Proceedings of the European Conference on Computer Vision (ECCV'10)*. 142–155.

H. Wu, Y.-S. Wang, K.-C. Feng, T.-T. Wong, T.-Y. Lee, and P.-A. Heng. 2010b. Resizing by symmetry-summarization. *ACM Trans. Graph.* 29, 6, 159:1–159:9.

J. Xiao, T. Fang, P. Tan, P. Zhao, E. Opek, and L. Quan. 2008. Image-based façade modeling. *ACM Trans. Graph.* 27, 5, 161:1–161:10.

C. Zach, A. Irschara, and H. Bischof. 2008. What can missing correspondences tell us about 3D structure and motion? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*. 1–8.

C. Zach, M. Klopschitz, and M. Pollefeys. 2010. Disambiguating visual relations using loop constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*. 1426–1433.

Q. Zheng, A. Sharf, G. Wan, Y. Li, N. J. Mitra, D. Cohen-Or, and B. Chen. 2010. Non-local scan consolidation for 3D urban scenes. *ACM Trans. Graph.* 29, 4, 94:1–94:9.

Y. Zheng, X. Chen, M.-M. Cheng, K. Zhou, S.-M. Hu, and N. J. Mitra. 2012. Interactive images: Cuboid proxies for smart image manipulation. *ACM Trans. Graph.* 31, 4, 99:1–99:11.

Received November 2012; revised June 2013; accepted August 2013