Multi-view Repetitive Structure Detection

Nianjuan Jiang  Ping Tan  Loong-Fah Cheong
National University of Singapore

Abstract

Symmetry, especially repetitive structures in architecture are universally demonstrated across countries and cultures. Existing detection methods mainly focus on the detection of planar patterns from a single image. It is difficult to apply them to detect repetitive structures in architecture, which abounds with non-planar 3D repetitive elements (such as balconies and windows) and curved surfaces. We study the repetitive structure detection problem from multiple images of such architecture. Our method jointly analyzes these images and a set of 3D points reconstructed from them by structure-from-motion algorithms. 3D points help to rectify geometric deformations and hypothesize possible lattice structures, while images provide denser color and texture information to evaluate and confirm these hypotheses. In the experiments, we compare our method with existing algorithm. We also show how our results might be used to assist image-based modeling.

1. Introduction

Repetitive structures are commonly observed in architecture. They can be utilized in architecture modeling to achieve data completion, refinement or compression as demonstrated in recent works [21, 31]. However, it is a difficult problem to detect these structures automatically.

Existing works on symmetry detection mainly focus on the detection of planar patterns from a single 2D image. There is a series of works, e.g. [14, 16, 11, 9, 22, 23, 12, 28], to categorize and detect symmetries. When the repetitive structure lies on a curved surface, the detection is complicated by the deformation of repetitive elements and their lattice structure. To handle this problem, Hays et al. [9] and Park et al. [22] iteratively rectify the surface and detect a lattice structure in the rectified surface. However, this simultaneous estimation of deformation and lattice structure leads to complicated optimization. It is also difficult to apply them to non-planar 3D repetitive elements.

Real buildings often contain 3D repetitive structures such as balconies and windows. They can lie on curved building façades, which makes the detection even harder. Some examples of such buildings are provided in Figure 1. Focusing on these challenging data, we study repetitive structure detection from multiple images of the same scene. We employ the structure-from-motion algorithm to reconstruct 3D point cloud from these images. There are two naïve ways for repetitive structure detection based on our input. First, we might rectify these images and detect repetitive pattern in the rectified picture with conventional methods. However, as shown at the top of Figure 1 (b), the non-planar repetitive elements (e.g. the red balconies) could make the rectified image asymmetric. Second, we might apply 3D symmetry detection methods, e.g. [24, 19, 5], to the reconstructed 3D points. However, our points are too sparse and noisy, as shown at the bottom of Figure 1 (b), to apply these methods, which require local geometric features such as surface curvatures.

Hence, we propose to jointly analyze the reconstructed points and the multi-view images for repetitive structure detection. We first identify repetitive 3D points according to their image appearance in multiple views. We then estimate the underlying surface of these points by assuming that they can be described by a ruled quadric model, and rectify it to a plane to facilitate the analysis. Translation and reflection symmetry are hypothesized from the 3D points in the rec-
tified surface, and then validated in the images to make it robust to noisy 3D points. The detected repetitive structure can help us to enhance the quality of the reconstructed 3D points, which can benefit image-based modeling works like [29, 30].

2. Related Works

Many symmetry detection algorithms have been proposed. Most of them [26, 14, 25, 15, 18, 11] focused on planar patterns. These methods can be regarded as local or global according to their methodologies. Local approaches like [25, 18, 15] extract a sparse set of corresponding features and hypothesize symmetry foci from pair-wise matches. These symmetry foci are then identified either via some voting schemes in a Hough transform fashion [25, 18] or exhaustive search in the parameter space [15]. Global approaches use autocorrelation [14], the Fourier transform [11], or co-occurrence matrices [26] for discovering periodic patterns. All these methods share a common disadvantage in that both the repetitive elements and the underlying surface of these elements are assumed to be mostly flat and frontoparallel in the image. Hence, they can hardly be applied to our data.

When a planar pattern is imaged from a slanted viewpoint, there is significant foreshortening effect. Cornelius and Loy [6] proposed a method to detect planar bilateral symmetry under such kind of perspective distortions. Wu et al. [28] rectified images according to vanishing points to facilitate repetitive structure detection. It is more challenging when the repetitive pattern lies on a curved surface, which causes spatially variant deformation. Hays et al. [9] iteratively rectified and estimated the topological lattice. This was further extended in [22] with the mean-shift belief propagation method to optimize the position of all lattice grids together. However, these methods require complicated optimization. Furthermore, they do not consider non-planar 3D repetitive elements, which are common in real buildings. In comparison, we utilize multiple images and 3D information from multi-view reconstructions for non-planar repetitive structure detection.

There are also a number of methods to detect symmetry in 3D data. Pauly et al. [24] and Mitra et al. [19] estimated symmetry of dense laser scanned 3D data by analyzing its geometric signatures such as curvatures and tangent coordinate systems. In comparison, Bokeloh et al. [2] designed a novel ‘line features’ for symmetry detection. In a recent work, Bokeloh et al. [3] further applied the detected symmetries for inverse procedure modeling. Combes et al. [5] computed the symmetry plane of bilateral objects from laser scanned point clouds. Thrun and Wegbreit [27] searched for symmetries based on a hierarchical generate-and-test procedure. All these works require dense 3D point clouds for symmetry detection. Though we reconstruct 3D points from multi-view images, our data are much sparser and noisier, which makes these methods unsuitable. By utilizing rich texture information provided by multiple images, we can overcome the problem of sparse 3D points.

Our work is also related to methods that exploit repetitive structures to facilitate architecture modeling. Müller et al. [20] analyzed the window patterns on a façade plane to generate its detailed 3D model. Korah and Rasmussen [10] detected and removed occluding objects from images by repetitive pattern analysis to generate clean texture maps. Nan et al. [21] and Zheng et al. [31] employed interactive methods to identify repetitive structures in laser scanned points for architecture modeling.

3. Overview

Starting from multiple images of the same scene, we first apply structure-from-motion algorithms [8, 13] to obtain a cloud of 3D points. Typically, we get about 50,000 visible 3D points in each image (of resolution 1200 × 800). An example of this reconstruction is shown in Figure 1 (b). We first identify multiple groups of repetitive 3D points to estimate the underlying curved surface (See Section 4). We rectify this surface to a plane to eliminate the geometric deformation of the underlying lattice structure. The appearance variation of repetitive elements is implicitly handled by the SIFT feature descriptor which is more robust to variations than the NCC approach in [9, 22] and by the availability of multi-views. We identify a lattice structure for each group of repetitive points. We then cluster and merge these results and report the most dominant one for each surface (See Section 5). Detected repetitive structure might be applied to clean up the reconstructed point cloud as in [21], which is helpful for image-based modeling applications (See Section 6). Experiments on real data and comparison with existing work [22] are provided in Section 7.

4. Geometry Rectification

We begin by identifying 3D points with similar image appearance and estimating the underlying curved surface. The surface is then rectified to remove geometric deformations so as to facilitate lattice detection.
4.1. Repetitive Points Identification

We use the SIFT features [17] already extracted for 3D reconstruction. We associate each reconstructed 3D point with these features in multiple images where it is visible. We exhaustively check all pairs of SIFT descriptors from different 3D points. Repetitive points are identified if the angle between their descriptors is smaller than a threshold $\theta_1$ (20 degrees, in our implementation).

We consider the matching of repetitive points as an equivalence relationship. In other words, if two points both match with a third point, we also consider these two points as matched repetitive points. At the end of this step, we have repetitive 3D points in different groups according to their 3D points. Repetitive points are shown in Figure 2. Points from the same group are marked in the same color.

4.2. Structure Estimation

These repetitive points could lie on a curved surface, which causes geometric deformation of the lattice structure and complicates the detection. We recover and rectify this surface to facilitate the detection. We assume this surface is either a plane or a ruled quadric, which is true for most real buildings.

We apply RANSAC [7] to fit either a quadric or a plane to each group of 3D points and select the model with most inliers. For every nine sampled points that pass the degeneracy testing, we fit a quadric using the normalized linear solution. The quadric is further converted to its canonical form and classified into ruled quadric, degenerate quadric and general quadric based on rough rank estimation. We keep only ruled quadrics to describe points on curved surfaces, and the rest will be described using planes.

Different point groups (e.g. different corners on the repetitive balconies) often lie on similar surfaces that differ from each other by a small translation. We cluster these groups together. Ruled quadrics and planes are clustered separately. For this clustering, we simply stack all the 16 elements in $Q$ or the normal of the planes to characterize a group. Two groups are clustered together if their 3D point constellations are close in space and their normalized parametric model vectors span an angle less than $\theta_2$. (We fix it at 2 degrees in our implementation.)

The surface fitting can then be refined from multiple groups. Suppose the groups $g_1, g_2, \cdots, g_N$ are clustered together. We refine the surface $\mathcal{S}$, a ruled quadric or a plane, by minimizing the following objective function

$$\sum_{i=1}^{N} \sum_{p \in g_i} R(p - d_i, \mathcal{S}).$$

Here, $R(x, \mathcal{S})$ is the algebraic distance between a 3D point $x$ to the surface $\mathcal{S}$. $d_i$ is a translation in 3D space, which allows the surface of different groups to differ from each other by a translation. This minimization is solved in an iterative fashion. In each iteration, we first fix all $d_i$ to estimate $\mathcal{S}$ and then fix $\mathcal{S}$ to estimate $d_i$ for each group respectively. Both estimations only involve a linear equation and the whole process converges quickly. We begin this iterative fitting by letting $d_i$ equal to zero. Some surface fitting results are illustrated in Figure 3. These surfaces are then rectified to a plane to facilitate the analysis.

5. Joint Repetitive Structure Detection

We jointly analyze the reconstructed point cloud and multi-view images for repetitive structure detection. We use 3D points to initialize repetitive structure hypotheses and verify them in images. Note that after we rectify the curved surface, like in the case of the Rome Colosseum example, the original rotational symmetry becomes a translational symmetry. Hence, we only consider points that are related by translations or reflections in the rectified surface. This treatment of rotational symmetries is more general than that in [24], which estimates a 3D rotation axis and an angular interval and cannot handle elliptic cylinders like the Rome Colosseum example.

5.1. Translational Lattice Detection

We first detect the underlying lattice for each group of points. We then consolidate these results to choose the most reliable parametric model for all groups $g_1, g_2, \cdots, g_N$ that share the same surface $\mathcal{S}$.

Lattice initialization The 2D lattice structure is characterized by its two basis vectors. In the rectified surface, we check all pairs of repetitive points within a group, and compute a translation between each pair. A naive lattice detection is to select the highest two local peaks in the histogram of these translations as the basis vectors. However, its performance is poor because the reconstructed 3D points are quite sparse and noisy. We treat these local peaks as candidate basis vectors and verify them according to the images as detailed below. An example histogram is provided in the first row of Figure 4.

Lattice validation To verify a basis vector, we select a 3D point as reference. Multiple grid points can be predicted
Figure 4. Top row: the histogram of pairwise translations and the detected lattice from this raw histogram. Bottom row: the image validation score of pairwise translations and the detected lattice from this score. In both rows, the two selected lattice basis vectors are circled in red. Note that in the original histogram space, one of the correct basis vectors has very low vote.

Figure 5. The red cross is a reference point. Blue and white circles indicate valid and invalid grid points. The SIFT descriptor of each point is obtained from its own most frontoparallel image. The two green crosses are the two farthest reconstructed points on the grid, which help to decide the width of the grid.


Hence, instead of using the number of valid grid points, we use the ratio between this number and the total number of grid points within a boundary. The sum of this ratio over all reference points is defined as the image validation score of a basis vector. To decide this boundary, as illustrated in Figure 5, we search for two farthest reconstructed on-grid 3D points located on both sides of the reference point in the feature group. We consider a point to be on-grid if the distance between it and certain grid point is smaller than a threshold $T_1$ (10% of the basis vector length). Starting from these two initial points, we move them away from the reference point along the line until a significant portion $T_2$ (50% in our experiments) of the grid points within them are invalid. We then trim all the invalid grid points at both ends to obtain the grid’s boundary.

After calculating an image validation score for all candidate basis vectors as shown in the second row of Figure 4, we can choose two of them to form the lattice structure. We sort these vectors in descending order of their lengths, and analyze them from top to bottom of the queue one by one. A vector with longer length is discarded, if it can be represented as an integer combination of the rest of the queue. If two vectors are along the same direction, we only keep the one with higher score. Finally, we select two vectors with highest score from the remaining ones.

**Lattice boundary estimation** Once the two basis vectors are selected, we proceed to generate the lattice grids. The main challenge here is to decide a precise boundary of the lattice. We start from a 3D point and expand the grid by one row/column at a time, as shown in Figure 6. If the proportion of invalid points in an expanded row/column exceeds the significance threshold $T_2$, we stop and try to expand along the other directions. The lattice is finalized once all four directions cannot be expanded.

**Lattice consolidation** In real buildings, all the repetitive point groups on the same curved surface (e.g., different groups of repetitive corners on balconies) share the same lattice structure. Hence, we can consolidate the detection among these groups and generate one final result for each surface. We form multiple clusters of lattice structures. Two lattices are clustered together if the difference between their translation vectors is smaller than the threshold $T_1$. Among

---

1 We use a looser threshold than the one in the detection of repetitive 3D points.
these clusters, we only keep the one with the largest number of lattices. An example of this consolidation is shown in Figure 7, where the biggest four clusters are shown. The number of grids in a cluster is shown at the upper-left corner. For this example, we finally choose the right most cluster.

5.2. Local Reflection Symmetry Detection

After the lattice detection, we can look for local reflection symmetries on the lattices. Like [28], we only consider reflection symmetries in the ‘vertical’ direction, which is the direction of one of the two lattice basis vectors. We choose the one that is closer to the up direction of the input images.

Reflection axis estimation

We exhaustively check all the vertical axes in the rectified surface. All 3D points used to fit this surface are used to vote for the right axis. For each candidate axis, we compare the SIFT descriptor of a 3D point with that of its mirrored point according to the axis. This pair of points is considered as valid if the angle between their SIFT descriptors is smaller than \(2\theta_1\). Again, both descriptors are obtained from their most frontoparallel images. We build a histogram of the number of valid pairs for all axes. If only a single dominant peak is found in this histogram (i.e. the second highest peak is lower than half of the highest one), we choose it as the reflection axis. Otherwise, there exist multiple valid axes. The horizontal interval between two neighboring axes should be the same as that between two lattice points [28]. Hence, we fold the original histogram according to this interval, i.e. \(\hat{H}(k) = \sum_{i=k-T}^{k} H(i)\), where \(T\) is the interval, and find the strongest peak in the folded histogram to locate all these axes.

Symmetry boundary estimation

For each detected symmetry axis, we set the boundary of the associated region as the bounding box of its valid point pairs. In the case of multiple repetitive symmetry axes, we compute a common boundary for all axes in the ‘vertical’ direction from their valid point pairs. Their width in the other direction is the same as the interval between axes.

6. Point Clouds Consolidation

Once the repetitive structure is identified, we can use it to enhance the point cloud like [21] to facilitate image-based modeling. Here, we only apply the translational symmetry to demonstrate this idea. We extract and align multiple blocks of 3D points according to the underlying lattice to generate more complete and denser results. All 3D points projected within a lattice cell form a block. Multiple blocks are extracted at different cells and aligned according to the lattice periodicity. We further apply the iterative closest point (ICP) algorithm [1] to refine the registration. Plane fitting and outlier removal could be applied subsequently. Figure 10 shows some point clouds before and after consolidation.

7. Experiments

We evaluated our method on multiple buildings with 3D non-planar repetitive elements. Some of the examples are provided in Figure 8. We used about 15 input images for
Figure 8. Results of repetitive structure detection. (a) and (b) are the left most and right most views of all input images. The detected repetitive points are overlaid on the image (the same group of repetitive points are visualized in the same color). (c) and (d) show the detected lattice and local reflection symmetry respectively.

As a rough average, our 3D reconstruction algorithm reconstructed about 100,000 points for each example and 50,000 visible points for each image, which is quite sparse compared with the image resolution, about $1200 \times 800$ pixels in our experiments. The first two columns of Figure 8 show the left most and right most views of each building. The matched repetitive points are overlaid in these images, where points of the same color are from the same group. Typically, our program identified about 50 groups of repetitive points on each example. The detected lattice is shown in (c). For each example, we provide a few (1-3) different lattice that share the same basis vectors. In these examples (especially the first three rows), the repetitive elements are clearly non-planar. Yet our algorithm still correctly identified the lattice. The detected reflection symmetry is visualized in (d). The reflection axis is shown in green and the boundary is indicated by a yellow box. Note that our method works for images with multiple buildings, see Figure 9. Furthermore, we apply RANSAC iteratively for surface detection within a repetitive point group, repetitive structure on multiple similar buildings can also be detected, e.g. the first example in Figure 12.

Additional results are reported in Figure 12. In these examples, (a) shows one of the input image with detected repetitive points. (b) is the estimated surfaces. To demon-
Figure 11. The first row is the lattice detected by our method, and the second row is that by [22]. (a) Both methods detected correct lattice. (b) Our method detected partial lattice and [22] detected wrong lattice. (c) Our method outperformed [22] in this case.

strate the potential in image-based modeling, we further manually create a mesh for one repetitive element according to its consolidated 3D point cloud \(^3\). This mesh is then tiled over the lattice to generate the result shown in (d).

**Comparison with [22]** We compared our method with [22] on 16 different scenes with 373 images in total. We used the code provided by the authors. Some detection results from both methods are provided in Figure 11. It is clear that [22] tends to fail when the repetitive element resides on a non-planar surface. We consider the detection a failure if a) no lattice is detected. b) wrong basis vectors are detected. or c) the detected lattice region is less than 30% of the actual one in the image. We evaluate the performance of both methods by two different counting rules: 1) use each image as a data sample. 2) use each sequence as a data sample, and score according to the best result in the sequence. Our method succeeds in 81 and 100 percent of these images, respectively. In comparison, the method in [22] can only handle 22 and 75 percent of the data, respectively\(^4\). We believe the strength of our method stems from the joint analysis of multi-view images and the reconstructed 3D points.

**Point cloud consolidation** To exemplify the point cloud consolidation, we provide examples before and after consolidation in Figure 10. It is clear that the original reconstructed points are much sparser with many holes. In comparison, the consolidated results capture the shape detail much better. These examples correspond to the buildings in the third row in Figure 12 and the first row in Figure 8 respectively. Please refer to their pictures to verify the geometric details.

---

1Note that this element could be automatically generated by applying methods like [4].

2Refer to supplementary data for detailed results.

8. Conclusion and Discussion

We present a method to detect architecture symmetries from multi-view images. Our method jointly analyzes these images and a cloud of 3D points reconstructed from them. We fit quadrics or planes to these 3D points to initialize repetitive structure detections and verify these initializations according to images which contain dense color and texture information. The fitting works well most of the time, though it confuses shallowly curved surfaces with planes sometimes. Our method contains a number of thresholds, i.e. \(\theta_1, \theta_2, T_1, T_2\), to decide if two SIFT features are similar, and if two groups are close to each other. However, since we apply them on normalized data, these parameters are easy to set and they are all fixed in our experiments. Thus, we conclude that the use of multiple views and joint analysis of 2D and 3D information make the algorithm capable of handling challenging data.

**Acknowledgements**

We thank the anonymous reviewers for their valuable feedbacks and Han Shuchu for his technique assistance. This work was supported by the Singapore grant R-705-000-018-279, R-263-000-555-112 and R-263-000-620-112.

**References**

Figure 12. Additional results. (a) shows one of the input images with detected repetitive points (the same group of repetitive points are visualized in the same color). The underlying surface of these feature points is visualized in (b). (c) shows the estimated lattice. (d) is the 3D model of the surface.


