

Matching Many Identical Features of Planar Urban Facades Using Global Regularity

Eduardo B. Almeida
Brown University

eduardo_almeida@brown.edu

David B. Cooper
Brown University

david_cooper@brown.edu

Abstract—Reasonable computation and accurate camera calibration require matching many interest points over long baselines. This is a difficult problem requiring better solutions than presently exist for urban scenes involving large buildings containing many windows since windows in a facade all have the same texture and, therefore, cannot be distinguished from one another based solely on appearance. Hence, the usual approach to feature detection and matching, such as use of SIFT, does not work in these scenes. A novel algorithm is introduced to provide correspondences for multiple repeating feature patterns seen under significant viewpoint changes. Most existing appearance-based algorithms cannot handle highly repetitive textures due to the match location ambiguity. However, the target structure provides a rich set of repeating features to be matched and tracked across multiple views, thus potentially improving camera estimation accuracy. The proposed method also exploits the geometric structure of regular grids of repeating features on planar surfaces.

I. INTRODUCTION

Duplicated structures are ubiquitous in urban scenes and handling their inherent match ambiguity has been an important topic in computer vision. Scene structure duplicates usually consist of man-made objects and are commonly found in architectural scenes. Facades and windows are among the most common repeating elements in urban scenes, as seen in figure 2.

Replicated objects often cause issues in *structure from motion* (SFM), the problem of simultaneously estimating camera poses (motion) and 3-d points (scene structure) from 2-d images with correspondences. Duplicated objects often give rise to erroneous or ambiguous matches and their detection and disambiguation must rely on global consistency measures. For instance, consider a simple scene with two identical cereal boxes, box 1 and box 2, and a reference image sees only box 1. This scene configuration can illustrate two common possible ambiguities problems. Traditional matching, e.g. via SIFT [7], a broadly used local feature descriptor in SFM, may incorrectly associate the image objects if the sensed image sees only box 2 (*type I* ambiguity). SIFT will likely fail to provide correspondences from the boxes if the sensed image sees both of them, as it cannot disambiguate the nearly identical local descriptors (*type II* ambiguity). Note, *type I* ambiguity causes an error, whereas *type II* leads to missing matches. These issues have motivated work in scene disambiguation to determine the *correct associations*, i.e.

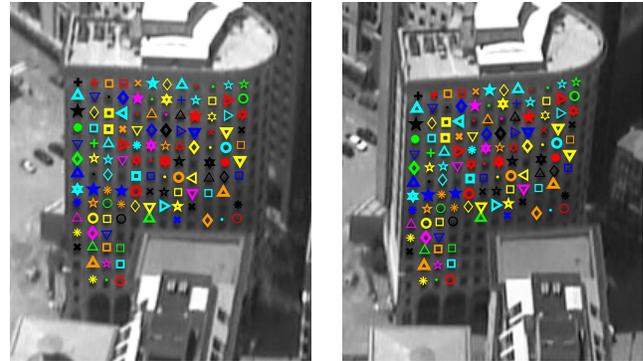


Figure 1. Typical example of the model-based pairwise matching method proposed in this paper. A building facade exhibits many windows with identical appearances duplicated in multiple locations in both views, leading to highly ambiguous correspondences due to repetition. Matches for over 100 repeating features are correctly disambiguated under partial facade occlusion with no calibration nor prior knowledge about the scene. Corresponding points are shown with identical markers.

matching features that correspond to the same 3-d points. Most existing work focus on *type I* ambiguity where repeating elements are *large* objects repeating only a few times in a scene. Progress has been made in terms of detection and matching in the presence of such objects [10], [8], [6], [2], [13], [3].

In contrast, the proposed method focus on match disambiguation for scenes with enhanced *type II* ambiguity, where multiple distinct repeating elements may exist, each one exhibiting a *massive* number of virtually identical features, as a scene with all the buildings in figure 2. The proposed procedure targets aerial views of scenes with tall buildings, where multiple facades may be visible at once and the common repeating elements are the windows of each facade, but it is not limited to such objects or imagery. No prior information about camera parameters are assumed.

Contributions: In summary, the main contributions of this work include: (1) proposing a novel lattice-based solution to a very ambiguous image correspondence problem on matching a large number of identical features from building facades, (2) enabling registration of the planar surfaces via homographies to achieve straight forward wide baseline matching and tracking via normalized cross-correlation, and to (3) improve SFM accuracy on urban scenes with a greatly enhanced number of matched features from facades.



Figure 2. Example of facades of large buildings from a single urban scene. All buildings present planar grid repeating features.

II. RELATED WORK

The majority of previous work focus on scene disambiguation of large objects under minor repetition, *i.e.*, repeating elements occur a few times in the scene and are often seen only once in an image (*type I ambiguity*). Specialized SFM algorithms for handling repeated objects in a scene have been proposed [13], [10], [3], [12], [1] and typically use SIFT matching for finding correspondences.

In contrast, the proposed method tackles repeating elements displaying *massive dense repetition*, such as hundreds of nearly identical features in a single image and proposes an approach to robustly match all of them under viewpoint changes assuming an they are organized in a flat lattice (see figure 1). The lattice grid detection and matching is robust to incomplete detected grids and also missing grid matches. Park *et al.* [9] detect a deformed lattice from repeated patterns in a single image, but does not propose stereo matching. Deformed lattice matching is usually achieved using spatial-temporal tracking assuming negligible inter-frame motion and do not handle significant baselines [5]. Schindler *et al.* [11] matches 3-d patterns from a pre-existing database of geolocated planar facades to newly detected facades exploiting the repeating nature of buildings to match the lattice rather than individual points.

Recent work of Liu and Liu [6] detects and associates facades on unconstrained aerial views and it is more directly related to this work. [6] achieves good results in detecting multiple facades per image and in associating detected facade regions in different viewpoints. However, the matched facade regions do not provide accurate local feature correspondences of the ambiguous repeating elements (window corners).

III. OVERVIEW

Appearance based matching algorithms typically fail to establish a unique correspondence between two views of a highly repetitive texture due to the fact that the appearance is densely duplicated at multiple locations in both images. Highly repetitive textures mostly occur on planar man-made objects, especially in urban scenes with repeating

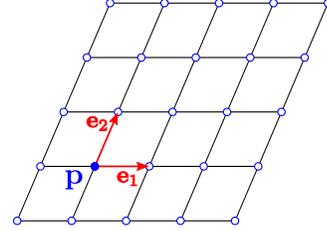


Figure 3. Definition of planar lattice: a repeating arrangement of points in space defined by a point p and two generator vectors, e_1 and e_2 , resulting in a regular tiling of the plane by parallelograms.

structure, *i.e.* a homogeneous facade of windows. However, by introducing a planar geometric constraint, the algorithm proposed in this paper establishes a correspondence between two views of the same repetitive structure by exploiting the global planar geometry to disambiguate features with identical appearance.

The proposed matching algorithm is outlined in figure 4. In a reference image, potential grid points are detected as features that have many surrounding replicas in their local neighborhood found via normalized cross-correlation, and are defined as grid seeds. After selecting one grid seed point, a larger neighborhood is searched for image locations with similar appearance (figure 4a). A Delaunay triangulation provides edge connectivity among these similar points (figure 4b) which may include outliers, points that are similar to the seed in appearance, but do not lie on the same planar object. After removing outliers from the original Delaunay triangulation (figure 4c), a regular lattice is inferred from the refined triangulation based on common edge lengths and orientations. See figure 3 for lattice definition. The estimated lattice grid points are the vertices of the refined triangulation and the two lattice edge directions are inferred from the refined triangulation common edge orientations (figure 4d). Unlike the triangulation, the lattice provides a genuinely regular topological structure to the grid points (compare figures 4c and 4d).

Grid points are matched individually in a sensed image via normalized cross-correlation while imposing the epipolar constraint. Note, the proposed approach is not limited to a choice of features and matching criteria, extending to any method that provides pairwise correspondences between repeating elements. Epipolar geometry is estimated from unambiguous matches found elsewhere. Essentially all the grid matches are still highly ambiguous (multiple match locations) when only considering appearance, due to the repetitive nature and density of the grid. Spatial regularity constraints of the lattice are used to disambiguate the point matches in two stages. First, constraints are applied to lattice lines, thus, simplifying the point correspondence ambiguity problem to disambiguating line matches (figure 4e). Finally, the intersection of two lattice lines must have a common corresponding point defined individually from their line match candidates, an important concept denoted intersection

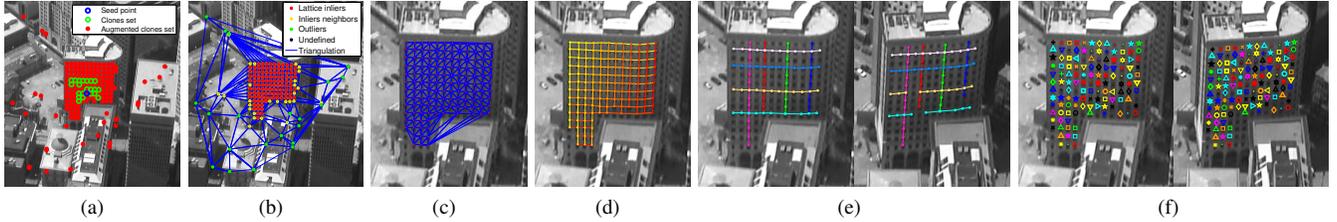


Figure 4. Overview of the planar grid matching pipeline. In a reference image, a grid seed point is found and similar features are detected as potential grid points (a); outliers are removed (b); inliers are triangulated (c); and a lattice is estimated from the triangulation structure (d). Global lattice spatial constraints are imposed on point matches and the problem of matching ambiguous grid points from the reference to the sensed image is reduced to a line matching problem (e). The lattice points are matched one line at a time in images taken from different viewpoints establishing correspondences (f).

consistency that ensures every grid point has a common correspondence from the two lines it belongs to. This concludes the model-based disambiguation pipeline, resulting in disambiguated matches (figure 4f).

After processing a lattice estimated from a seed point, a new seed is selected to process other lattices of the same image and this new seed does not belong to a previously learned or analyzed grid. The system only returns a corresponding lattice when both the learning and matching procedures succeed in estimating well-supported structures conforming with a proper planar lattice. Thus, corresponding lattices must present simple canonical properties derived from the definition in figure 3, such as line parallelism, no distinct lines intersect more than once, every node has no more than two neighbors per direction, the grid points are not all collinear and the correspondences satisfy a planar homography. These sanity checks prevent erroneous matches arising from false seed points or from falsely-detected lattices.

Limitations: The proposed method does not provide a correspondence for lattices of very skewed facades where the grid corners are starting to merge due to the oblique viewing angles, and also for camera configurations where epipolar lines and lattice lines coincide leading to multiple possible solutions. The proposed approach works well for large lattices and the smallest matched lattices have in average six to eight grid points.

A. Motivation for modeling planar surfaces

Planar geometry is very common in urban scenes and, under viewpoint changes, it preserves

- 1: the collinearity of features, and
- 2: the relative ordering of a set of collinear features.

These preserved properties are exploited to match a rich set of strong essentially identical features that are too ambiguous to match individually without a region model. The matching using a planar model, illustrated in figure 5b, succeeds for an entire grid of points and justifies the use of a distinct plane-based matching method tailored for this type of object.

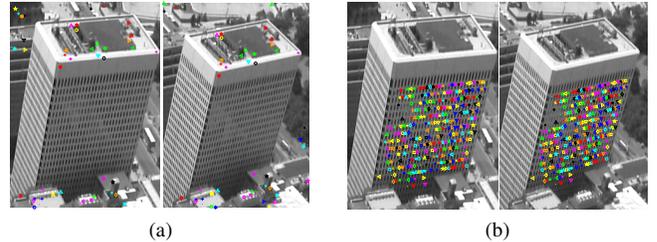


Figure 5. Matching results for highly repetitive urban scenes features. (a) Model-free matching using SIFT features. (b) Matching using proposed planar model of this paper.

IV. GRID POINTS DETECTION

A *grid* denotes a collection of nearly identical feature points called *grid points* or *clones* originating from a planar lattice in a 3-d scene. The clones in the planar grid lattice are evenly distributed displaying evident collinear subsets. Any clone can be a *seed* to a procedure that detects the grid for which it is a point.

A. Finding grid seeds

In order to find a corner that is a potential grid seed, normalized cross-correlation is used to correlate every corner point feature (11×11 patch) over a square region around itself of size 10% of the image height (72×72 in experiments). Matches are denoted *clones* (see figures 4a and 6). Points with a high number of clones are denoted *grid seeds* of an underlying grid structure. Note, only one seed is used to estimate a lattice.

B. Augmented clones set

A grid seed is a point that presents a large number of clones in its local neighborhood. In order to find the entire associated grid, a new correlation search is performed for the seed only, this time in a larger square neighborhood (50% of image height). The matches are then points that compose a planar grid in addition to outliers. The new set is denoted the *augmented clones set* and includes potentially more clones than the original clones set (see figure 6). The augmented clones set has outliers, is unordered and may miss some grid points. The method for robustly detecting the underlying true grid structure is explained next.

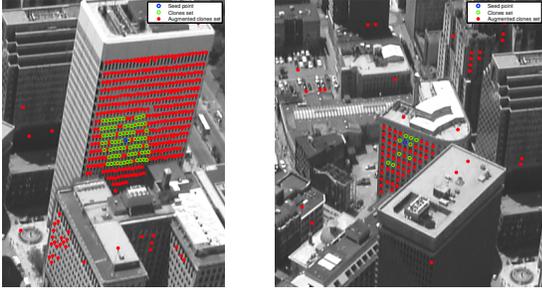


Figure 6. Illustration of two grids with grid seeds and clone sets. The grid seed points are in blue, their clones, composed of nearly identical features, are in green, and the augmented clones sets are in red.

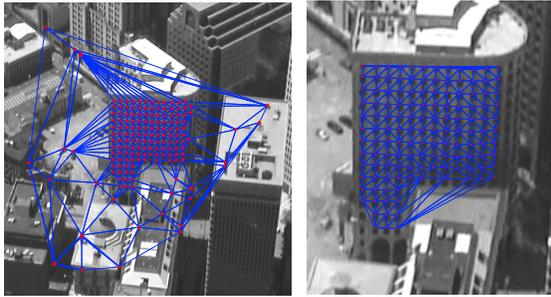


Figure 7. *Left*: A Delaunay triangulation of the augmented clone set of a grid seed showing potential grid points and some outliers. *Right*: triangulation of estimated inliers.

C. Lattice model

A Delaunay triangulation \mathcal{D}_t of the augmented clones set is computed and there is likely outliers present in the set, as in figure 7. On inlier regions, the edges of \mathcal{D}_t are often periodic as the lattice and therefore predictable, however this is not guaranteed since a Delaunay triangulation can be sensitive to some lattice configurations and present multiple periodic patterns. The proposed grid detection method is robust to a few distinct regular patterns. Near outliers, the triangulation edges are usually irregular. Thus, outliers are filtered out by learning the periodic patterns of \mathcal{D}_t and removing points that do not fit these patterns.

Lattice feature: The learning of the lattice structure is based on the feature vector

$$\mathbf{E} = \begin{pmatrix} \theta \\ l \end{pmatrix} \quad (1)$$

composed of the orientation θ and the length l of an edge of \mathcal{D}_t . The orientation angle and the length of an edge \mathbf{e} , given by the vector $\mathbf{e} = (e_x, e_y)^\top$, are illustrated in figure 8 and defined as

$$\theta = \arctan \frac{e_y}{e_x}, \quad (2)$$

$$l = \sqrt{e_x^2 + e_y^2}. \quad (3)$$

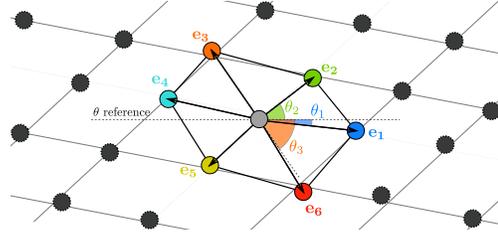


Figure 8. Pictorial representation of feature vectors of Delaunay triangulation edges of a grid lattice point. Six edge vectors are shown: $\mathbf{e}_1, \dots, \mathbf{e}_6$. The orientation components are displayed for the first three edges as θ_1, θ_2 and θ_3 and lengths are represented by the arrow sizes. By definition, the orientation feature for \mathbf{e}_1 and \mathbf{e}_4 are approximately the same and similarly for others.

Therefore, $-\pi/2 \leq \theta \leq \pi/2$. Edge vectors for neighbors that are in approximately opposing directions, e.g., Θ_e and $\pi + \Theta_e$, are represented by a single orientation.

Learning the lattice: In order to derive the lattice structure from the triangulation \mathcal{D}_t , assume the distribution of the edge feature vectors \mathbf{E} follows a 2-d mixture of Gaussians density function. The parameters are learned incrementally according to update equations based on [4] where samples are weighted equally.

The truly periodic features must be very common and produce strong Gaussian components, while the outliers are normally random and generate none. If there are multiple periodic patterns on \mathcal{D}_t , they are also learned.

D. Iterative outlier removal

Given the learned mixture of 2-d Gaussians, each vertex \mathbf{v} in \mathcal{D}_t is tested to check if all its edges are in accordance with the mixture. A discriminant value Z_e is computed for each edge \mathbf{e} of \mathbf{v} as the minimum of the Mahalanobis distances of the edge feature to each 2-d mixture component. The discriminants are used in an iterative pruning procedure that finds and removes highly inconsistent outlier vertices a few at a time, update \mathcal{D}_t and repeat.

An outlier on the triangulation interior affects the triangulation on its surroundings potentially causing any of its neighbors to also appear as an outlier. The proposed procedure copes with this problem to prevent an outlier from iteratively spreading its outlier condition outward as \mathcal{D}_t is updated, an issue denoted *outlier proliferation*. The solution is provided below.

Finding nongrid outliers: At each iteration, vertices such that at least 3 of their edges agree with the Gaussians mixture by having a small discriminant, i.e.,

$$Z_e < 2.5, \quad (4)$$

are considered inliers. Vertices at the borders and corners of the lattice need special treatment as some of their edges differ from the edges of interior vertices. Thus, neighbors of the inliers vertices may also be considered inliers and then are protected from removal at each iteration. The protected inlier neighbors (yellow nodes in figure 9) are the ones associated

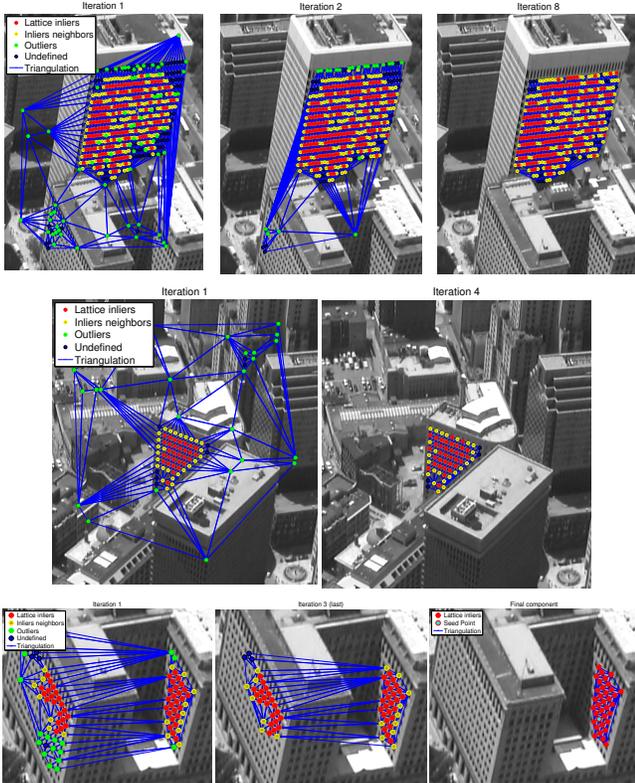


Figure 9. Iterative outlier detection and removal based on learned lattice.

with edges such that their discriminant is less than 2.5. This method preserves border vertices neighboring inliers and also avoids outlier proliferation. Remaining vertices are *nongrid outlier candidates* and are not promptly discarded, but must satisfy an inconsistency test to be considered an outlier. A candidate is an outlier if 60% of its edges discriminants (Mahalanobis distances) are greater than 5, *i.e.*,

$$Z_e > 5, \quad (5)$$

which means the edges are truly abnormal with respect to the learned model, as often observed on outliers (see figure 9).

Figure 9 displays the iterative outlier removal method, showing inliers, outliers and the triangulations. Outlier candidates that are not deemed to be estimated outliers are labeled “undefined”.

Termination condition: After each outlier removal iteration, the structure of \mathcal{D}_t is locally rearranged with the outliers removed and the procedure repeats until no outlier is left, as shown in figure 9, resulting in estimated inlier triangulations as in figure 10. Note that nearby identical facades may exist causing multiple grids to arise, as in the bottom of figure 9. In order to remove similar nearby grids, consider ignoring all edges not in accordance with the learned model as in equation (5), and keeping only nodes connected to the seed via remaining valid edges.

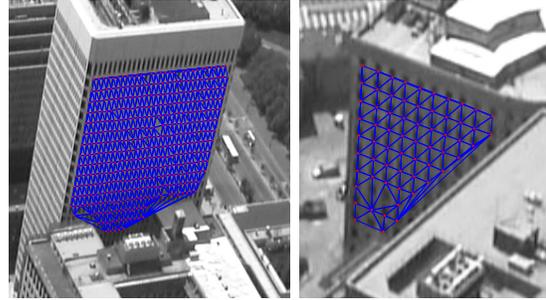


Figure 10. Examples of Delaunay triangulations of estimated grid points, shown as red dots. The triangulations (blue edges) depict the structure of the lattice.

E. Defining grid main directions

The main directions are any two directions in the grid where a lattice can be defined by connecting sets of collinear points parallel to each other. The directions are taken from the orientation components of the Gaussian mixtures learned in section IV-C. There are multiple possibilities for the choice of directions, as can be seen in figure 10. The main directions are chosen as two learned orientations such that their relative angle is close to 90 degrees (in the image).

V. GRID LATTICE ESTIMATION

Given estimated grid points, they must be connected in a consistent way in order to form a 2-d lattice, which consists basically of finding straight line connections along two main directions. Lattice lines must be estimated since the grid points alone are not useful to ease image matching disambiguation. As discussed in section III-A, matching is easier to disambiguate when constrained to a model and a planar surface.

To estimate lines from grid points, adjacent vertices are first connected in the two main grid directions, forming line segments (section V-A). A single grid line can be broken into multiple segments if there are missing vertices or local statistical variability. Second, the segments are iteratively merged into complete grid lines (section V-C).

A. Connecting vertices into line segments

In order to connect points into line segments in one of the main directions v , as in figure 11, every point is visited once. Given the first visited point p_1 , find its surrounding points according to \mathcal{D}_t and assign the neighbors along the direction v to be the one or two which are roughly in such orientation. This process iteratively repeats for the neighbors of p_1 and so on, growing it into a line segment of roughly collinear points, and is repeated for other points and for the other main direction. Results are shown in figure 11.

B. Defining line segments neighborhood

The line segments provide a rough structure of the grid that is enough to find line-to-line neighborhoods. Given a line segment s_l in one direction, the two neighbor parallel

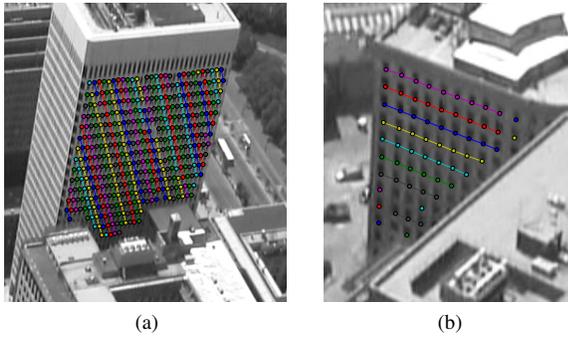


Figure 11. Line segments estimated for two grids of windows. Segments are presented with random colors for one grid direction. A single grid line may split into multiple segments and their breaking points are due to missing features, missing connectivity or small statistical deviations from learned orientations.

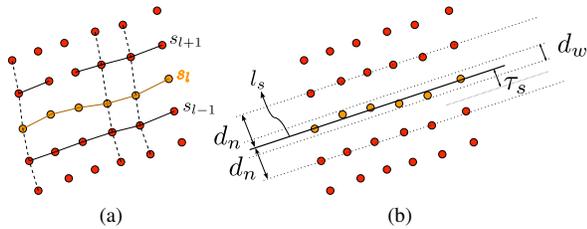


Figure 12. (a) Parallel neighbors of a lattice line segment s_l and three segments on the other lattice direction (dashed). (b) Distance threshold τ_s for points to merge into fit line l_s .

segments, s_{l-1} and s_{l+1} , can be found by traversing through the other main direction, as shown in figure 12a.

C. Merging line segments

Once all segments and their neighbors are established, they are merged to form longer and complete grid lines. The merging is based on distances from points to lines. Let s be line segment. A line is fit using SVD to the points in s , resulting in a line l_s . The distances from the points in s to l_s are computed and their median is the *within-line* distance, $d_w(s)$. The distances between neighbors of s and l_s are also computed and their median is the *neighbor-line* distance, $d_n(s)$. These distances are illustrated in figure 12b and their mean is a threshold τ_s ,

$$\tau_s = \frac{d_w(s) + d_n(s)}{2}, \quad (6)$$

representing the half-distance between a line and its neighbors that is used to determine whether a point should merge into l_s . This process runs from the largest to the smallest segments incrementally merging them into larger ones until nothing can be combined anymore.

Singleton-point segments: Segments with a single point can be naturally assigned to merge with other larger segments of the same underlying grid line via the proposed merging method. They may borrow line slopes from neighbor segments if necessary for merging. Near the lower left corner of figure 11b, two grid lines have only two singleton-point segments, but the lines are successfully learned (figure 13).

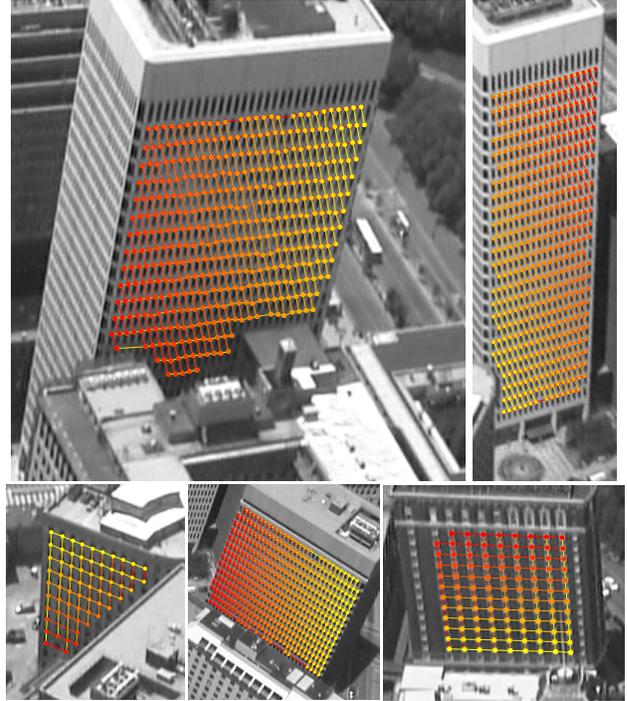


Figure 13. Examples of estimated 2-d planar grid lattices found in images from figure 2. Note, the method is robust to small distances between the grid lines.

D. Organizing lines into a 2-d lattice

The iterative segment merging process of section V-C is performed once in each main direction of the grid providing complete lines. The lines can be straightforwardly ordered given the estimated parallel line neighborhoods (section V-B). The result is an estimated 2-d lattice, as shown in figure 13, where the ordering of lines is color coded. In addition, the ordering of points in each line is illustrated by connecting adjacent points with a straight line segment. Note, the merging is robust to sequences of missing grid points.

VI. GRID MATCHING

After a grid lattice has been established, the next step is to match it on a sensed image enforcing global properties that are invariant under viewpoint changes to disambiguate the matching, as motivated in section III-A.

A. Match grid points independently

Individual grid points are first matched from a reference to a sensed image using traditional normalized cross-correlation procedure, retaining all high correlation peaks near epipolar lines. The epipolar geometry is estimated from other nongrid matched points from the image pair. The average number of matches per grid point is high due to periodicity of the grid (see figures 14 and 15 for details).

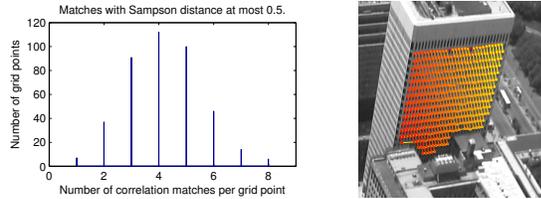


Figure 14. Histogram of the number of correlation matches on a sensed image (not shown) of repeating grid points displayed in the reference image on the *right*.

B. Reduce ambiguities by matching grid lines

Given that collinearity and ordering of grid points on a plane are preserved under viewpoint change (see section III-A), these constraints are used to rule out wrong matches. Given all matches of all grid points, it is likely that there is only one matching configuration among all possible combinations that preserves the topology of the entire estimated grid. However, trying every possible combination is prohibitive. Match ambiguity is reduced by several orders of magnitude by first constraining matches of collinear points taken from the grid lattice.

Three constraints are implemented to reduce ambiguities: *collinearity*, *ordering* and *spacing*. Points from a planar grid lattice line in a reference image must appear in the same relative order in the sensed image, in addition to being collinear. Spacing denotes the distances between *adjacent* grid line points. The spacing of the grid points and their matches is considered *locally* invariant under viewpoint changes under assumed approximately affine projection of building facades that are away from the camera. Spacing constraint is more relevant to disambiguate small lines. The tolerance level for the *collinearity constraint* is modeled using τ_s from equation (6). The *spacing constraint* is performed by comparing ratios of distances among neighbor points. In addition to relative spacing ratios, the absolute distances of matching neighbor points is only allowed to change by a factor of 1.5 between adjacent images.

Applying such constraints to points of a grid line, essentially eliminates match ambiguity. This constrained line matching robustly handles occlusion by tolerating missing grid points and skipping unmatched ones, as illustrated in figure 16. The bottom row of figure 17 provides matching results under occlusion. Multiple line matches may still exist after applying the constrains, yet affecting mainly short grid lines. When a single grid line has multiple line matches, only the ones with the highest number of points are kept for further disambiguation discussed in section VI-C.

C. Topology-preserving incremental line matching

The ambiguity of line matches seen in figure 16a is smaller than is the ambiguity of matching individual points (*c.f.* figure 15).

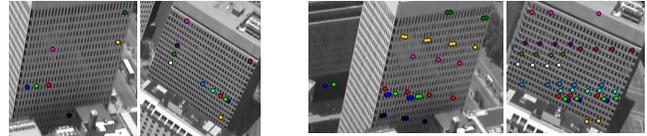


Figure 15. Examples of grid point matches computed individually. *Left*: arbitrary feature points, where some are collinear. *Right*: multiple matches of previous points in different views showing that collinearity alone does not fully disambiguate line matchings.

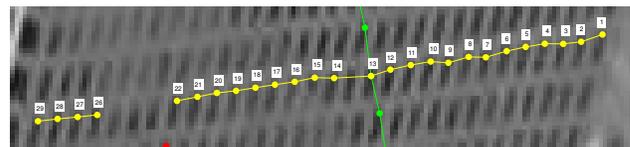
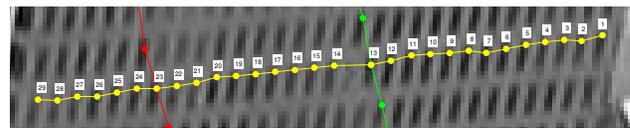
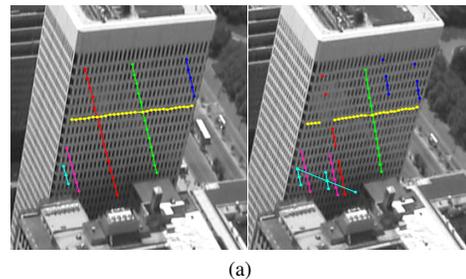


Figure 16. Illustration of grid line matching. (a) Aerial stereo pair showing grid lines in different colors (left) and their associated line match candidates (right) with matching colors. (b) Detail of the yellow horizontal grid line on the left image and (c) detail of its matching line on the right image. The numbers uniquely represent a line point and its match showing that matches are pointwise correct despite missing grid points and missing matches.

Grid topology is enforced using line intersections. The matching process starts by matching a long grid line using procedure in section VI-B. Longer lines are often unambiguous. Then, a second long line in the other direction is matched and its intersection with the first must be consistent. *Intersection consistency* is a very important concept implying that when two grid lines intersect at a point, their matching lines must also intersect at a common point. The matching expansion continues trying to add one line at a time to the matched grid, alternating between the two main directions. As new line matches are being incorporated, they must conform, in the context of intersections, with all lines matched prior to them. Note, consistent intersections are very restrictive, essentially eliminating match ambiguity by basically yielding one coherent matching line or none, and ensuring grid topology is preserved on the sensed image. Resulting matched grids are shown in figures 17 and 18.

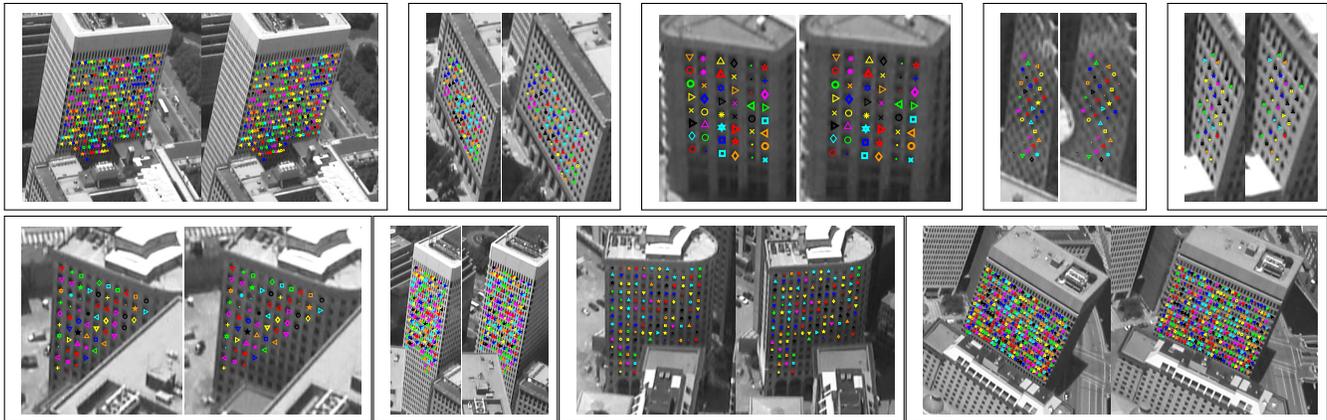


Figure 17. Examples of pairwise matches of grid features from building facades seen from distinct angles. The matches are estimated from proposed fully automatic procedure. Zoom in electronic version for details.

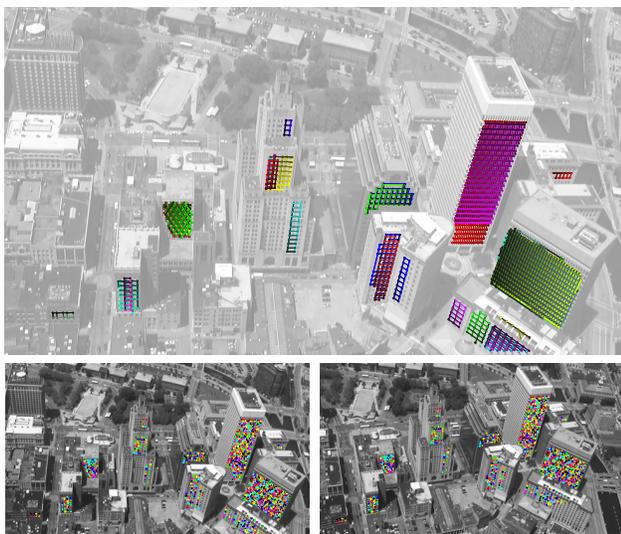


Figure 18. Multiple lattice correspondences automatically estimated using a single image pair of sizes 1280×720 . *Top*: a reference image and its estimated lattices that produced correspondences. A building facade may have multiple lattices, one for each distinct repeating corner from windows. *Bottom*: point correspondences of the multiple lattices (zoom in electronic version for details). The smallest matched lattice dimensions are 2×4 .

VII. CONCLUSION

The proposed feature matching method assumes the existence of grids of features with planar geometry in the scene and exploits this global spatial information to resolve a highly ambiguous correspondence problem. Unlike in previous work that only handles little repetition, common features that massively repeat, such as building windows found in urban scenes, are correctly detected and matched to one another by the proposed pipeline using a planar lattice model. The procedure is fully automatic and assumes no prior knowledge of the scene. The planar correspondences estimated here can be used to refine match location for piecewise planar regions under wide baselines by tracking planes among successive small-baseline images via homographies and can potentially improve the accuracy of SfM.

REFERENCES

- [1] D. Ceylan, N. J. Mitra, Y. Zheng, and M. Pauly. Coupled structure-from-motion and 3d symmetry detection for urban facades. *ACM TOG*, 33(1), 2014. 2
- [2] D. Hähnel, S. Thrun, B. Wegbreit, and W. Burgard. Towards lazy data association in slam. In *Robotics Research*, pages 421–431. Springer, 2005. 1
- [3] N. Jiang, P. Tan, and L.-F. Cheong. Seeing double without confusion: Structure-from-motion in highly ambiguous scenes. In *CVPR*, 2012. 1, 2
- [4] D.-S. Lee. Effective gaussian mixture learning for video background subtraction. *PAMI*, 27(5), 2005. 4
- [5] W.-C. Lin and Y. Liu. A lattice-based MRF model for dynamic near-regular texture tracking. *PAMI*, 29(5), 2007. 2
- [6] J. Liu and Y. Liu. Local regularity-driven city-scale facade detection from aerial images. In *CVPR*, 2014. 1, 2
- [7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2), 2004. 1
- [8] A. Martinović, M. Mathias, J. Weissenberg, and L. Van Gool. A three-layered approach to facade parsing. In *ECCV, LNCS*. 2012. 1
- [9] M. Park, K. Brocklehurst, R. T. Collins, and Y. Liu. Deformed lattice detection in real-world images using mean-shift belief propagation. *PAMI*, 2009. 2
- [10] R. Roberts, S. N. Sinha, R. Szeliski, and D. Steedly. Structure from motion for scenes with large duplicate structures. In *CVPR*, 2011. 1, 2
- [11] G. Schindler, P. Krishnamurthy, R. Lubliner, Y. Liu, and F. Dellaert. Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In *CVPR*, 2008. 2
- [12] K. Wilson and N. Snavely. Network principles for SfM: Disambiguating repeated structures with local context. In *ICCV*, 2013. 2
- [13] C. Zach, A. Irschara, and H. Bischof. What can missing correspondences tell us about 3d structure and motion? In *CVPR*, 2008. 1, 2