Detecting and Matching Repeated Patterns for Automatic Geo-tagging in Urban Environments

Conference Paper *in* Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition · July 2008

DOI: 10.1109/CVPR.2008.4587461 · Source: IEEE Xplore

5 authors, including:

Panchapagesan Krishnamurthy
Georgia Institute of Technology
3 PUBLICATIONS **209** CITATIONS

SEE PROFILE

Yanxi Liu
Pennsylvania State University
176 PUBLICATIONS **6,358** CITATIONS

SEE PROFILE

Frank Dellaert
Georgia Institute of Technology
247 PUBLICATIONS **17,169** CITATIONS

SEE PROFILE

Detecting and Matching Repeated Patterns for Automatic Geo-tagging in Urban Environments

Grant Schindler[1], Panchapagesan Krishnamurthy[1], Roberto Lublinerman[2], Yanxi Liu[2], Frank Dellaert[1]
[1]*Georgia Institute of Technology* [2]*Pennsylvania State University*
schindler@cc.gatech.edu, panchapagesan.k@gmail.com, rluble@psu.edu,
yanxi@cse.psu.edu, dellaert@cc.gatech.edu

Abstract

We present a novel method for automatically geo-tagging photographs of man-made environments via detection and matching of repeated patterns. Highly repetitive environments introduce numerous correspondence ambiguities and are problematic for traditional wide-baseline matching methods. Our method exploits the highly repetitive nature of urban environments, detecting multiple perspectively distorted periodic 2D patterns in an image and matching them to a 3D database of textured facades by reasoning about the underlying canonical forms of each pattern. Multiple 2D-to-3D pattern correspondences enable robust recovery of camera orientation and location. We demonstrate the success of this method in a large urban environment.

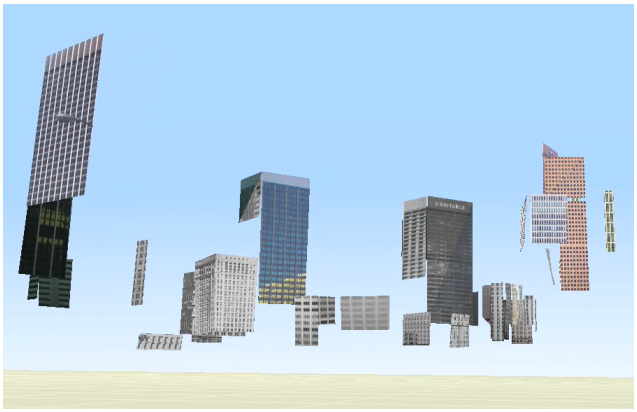

Figure 1. 3D Database of Textured Building Facades in a Major City. Given a textured 3D city model, we detect repeated patterns on building facades. By detecting the same repeated patterns in a new image, we automatically determine camera location and thus geo-tag the image.

1. Introduction

Automatic geo-tagging of photographs, i.e., tagging them with the location and possibly viewing direction from which they were taken, is fast becoming one of the most interesting challenges in computer vision. The proliferation of digital cameras and the genesis of web-sites and social networks for the photography enthusiast, such as Flickr or Picasa on the web, have already lead to a large number of photographs being manually geo-tagged by users themselves. However, this process is tedious and time-consuming, which is where an automated computer vision solution could offer relief.

Partly spawned by the ICCV Computer Vision Contest in 2005, a number of efforts have already resulted in impressive results. Recent work builds on seminal contributions in wide-baseline matching using affine-invariant descriptors [20, 13, 12, 18], which were not focused on geo-locating images but rather 3D reconstruction and image retrieval. Work explicitly focused on navigation is [4], where location recognition was achieved by matching line segments and their associated descriptors, followed by a geometric consistency check. A similar SIFT-feature-based approach is taken in [21] with a prototype system for image based localization in urban environments. Like them, we use a reference database of GPS-tagged images to enable geo-registering the query images.

However, SIFT features and other invariants are problematic in urban scenes due to the high degree of symmetry and repetition in these environments. In particular, these factors impede finding a geometrically consistent match between reference and query images, which is crucial for the geo-registration step. Several papers have taken advantage of the regular structure of urban scenes, e.g., [14, 15], but most notably [7], which successfully argues that man-made environments possess many regularities that should be regarded as assets rather than hindrances. These techniques deal neither with general symmetries, nor with the inevitable ambiguity in pose recovery results.

In this paper we propose a method which fully exploits the presence of so-called "wallpaper" patterns in the scene

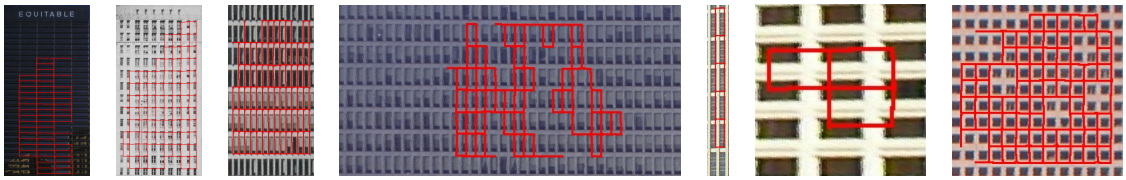

Figure 2. 3D Database Facades. We detect lattices (red) in textures that lie on geo-located 3D planar facades, thus generating 3D lattices with known positions in space. To geo-tag a new image, these 3D lattices must be matched with 2D lattices detected in the new image.

in order to automatically geo-register the image. Symmetric wallpaper patterns [8] are ubiquitous in man-made environments. However, repeated patterns viewed from different perspectives, such as photos of building facades in an urban scene, present computational difficulties when being matched to each other or to a 3D model directly. The primary challenge is its ambiguity in scale, unit patch and skewness of the matching patterns. A key contribution of our paper is the use of the skewed symmetry group theory developed by Liu and Collins [9] to resolve these ambiguities during matching. Moreover, the well understood symmetry properties of the wallpaper groups are used throughout all stages of our **automatic geo-tagging algorithm.**

At the highest level, our algorithm detects the perspectively distorted wallpaper patterns in a query image such as the one in Figure 3, and matches them against a reference database of geo-registered textures, as shown in Figure 1. In detail, the algorithm proceeds as follows:

- Off-line, create a 3D database of wallpaper patterns (Section 2)
- Detect the (perspectively distorted) repeated 2D patterns in an image (Section 3)
- For each pattern
  - match the 2D pattern to a 3D pattern in the database (Section 4)
  - recover a family of 3D camera poses (Section 5)
- Find poses common to each family (Section 6)

Below, we discuss each of these phases in the sections indicated above.

A Word on Notation

We use lowercase letters, e.g., t_1, t_2 to indicate vectors in 2D and uppercase T_1, T_2 for 3D vectors. Wherever we use rotation matrices we clarify the coordinate frames involved, e.g., R_a^b is a rotation *from* frame *a to* frame *b*. We specify cameras as a 3D location T_c and a rotation R_w^c from world to camera, i.e., the corresponding projective camera is $KR_w^c[I|-T_c]$, with K the calibration matrix

$$K = \begin{bmatrix} f & & u_0 \\ & f & v_0 \\ & & 1 \end{bmatrix}$$

i.e., assuming known aspect ratio and skew.

2. A 3D Database of Wallpaper Patterns

Our approach relies on a pre-existing database of geo-located planar facades (or other planar surfaces) that contain repeated patterns, associated with one of the 17 wallpaper groups, see e.g. [8, 9]. We observe that the facades of many structures in urban settings consist of multiple columns and rows of identical windows, yielding the appearance of repeated patterns. The defining characteristic of these patterns is the existence of two independent translations, hereafter called t_1 and t_2, and a *motif* that is repeated along these two translation directions. While there are 17 types of such patterns, the patterns most often occurring in urban scenes are those with rectangular lattices, where $t_1$and t_2 are aligned with the ground plane and gravity, respectively.

In our experiments, we use a database of textured, geo-registered, floating 3D quads that lie on building facades in a major city, as shown in Figure 1. This is an experimental proxy for the much larger texture-mapped 3D models of cities that are starting to appear in commercial products and are increasingly deployed on the web. Our model was built up from a collection of images using traditional structure from motion techniques.

Each 3D repeated pattern P^{3D} is stored as a 3D lattice, a rotation matrix R_l^w from lattice coordinates to world coordinates, and a set of labeled *motifs*. The latter make up an appearance model that is used for matching, and is explained in Section 4. The 3D lattice is given by two 3D vectors T_1 and T_2 corresponding to the 2D wallpaper group translations t_1 and t_2 for P^{3D}, and a 3D lattice origin L.

3. Lattice Detection

We use a variation of the RANSAC-based planar grouping method introduced in [16] to detect perspectively distorted lattices of feature points, which allows us to identify the two main translation vectors of the underlying repeated wallpaper pattern. Given an image of a scene containing multiple buildings, our goal is to detect all the repeated patterns (or lattices) present in the image. Recent work has led to a number of successful methods for detecting regular and near-regular textures in images [19, 16, 17, 6].

We adopt a feature-based method, using RANSAC [1]

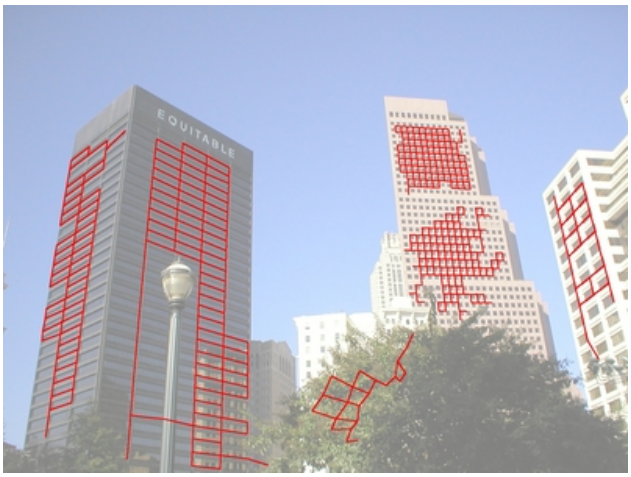

Figure 3. A photograph of an urban scene with several "wallpaper" patterns, a candidate for automatic geo-tagging via our approach. Here, we show the 2D lattices automatically detected in the image.

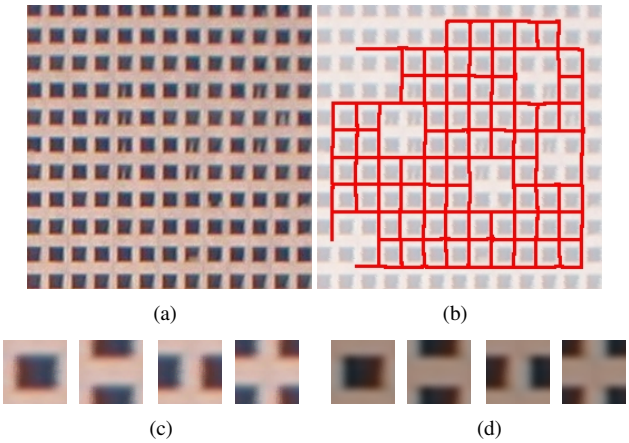

Figure 4. Database Motifs. Given the texture of a 3D quad in the database (a), we detect a lattice (b), and use it to generate a set of motifs (c) to represent lattice appearance. Note that we are able to recover the same basic motif set (d) from another image, which is easily matched to (c) amongst all database motifs.

to robustly identify lattices made up of SIFT features [10]. We first cluster the SIFT features into N groups based on their appearance descriptor. For each of these N sets of similar-looking features we randomly sample four points $\{a,b,c,d\}$ and compute the homography ${}_lH_i$ which maps these four points from image space into the lattice basis $\{(0,0),(1,0),(1,1),(0,1)\}$. We can now transform all remaining points from image space into their equivalent lattice positions via the homography ${}_lH_i$, and count as an inlier each point whose lattice space coordinates are within some threshold of an integer position (i,j). Indeed, if the four chosen points $\{a,b,c,d\}$ really do define a *motif* corresponding to a wallpaper group, we will find other repetitions of identical SIFT features. We further require that to be counted as an inlier, a point must be part of a connected component of other inliers in lattice space.

We end up with N detected lattices, one for each group of points, which we pare down to eliminate overlap and to eliminate lattices with too few inliers. We are left with some number less than N of non-overlapping, well-supported 2D lattices of points. See Figure 3 for an example of lattices detected in this manner.

4. 2D-to-3D Repeated Pattern Matching

We match the wallpaper patterns in the 3D database to their perspectively distorted 2D images using a canonical representation of lattice appearance called a *motif*, based on the skewed symmetry group theory set forth by Liu and Collins in [9]. To recover the motif of a repeated pattern we apply the algorithm from [9]. This amounts to finding the median value of every pixel in the repeated tiles of the detected lattice to form a *median tile,* and then translating, rotating, and reflecting this *median tile* back onto itself to find its symmetry properties. The canonical motif of a pattern is then represented by four 50×50-pixel images that reflect the inherent symmetries of the repeated pattern.

We first compute such a set of canonical motifs to represent the appearance of each textured 3D facade in the database, as illustrated in Figure 4. Note that each building facade in the database now has both a 3D lattice describing its *geometry* and a motif set describing its *appearance*.

The process for repeated patterns detected in the images involves coping with affine distortion. While we can remove any perspective distortion using the recovered vanishing point information from Section 3, the resulting rectified texture might still differ from its corresponding database texture by an affine transformation. In this case, the symmetry properties of a given repeated pattern may change, as illustrated in Figure 6. Again we use the algorithm in [9] to determine all potential symmetry groups of a pattern under affine deformation, and its corresponding set of motifs, which gives us an affine invariant to use in matching.

To establish a 2D-to-3D lattice correspondence, we use the motif recovered from the image to match against similar motifs in the 3D database. The measure of similarity between any two lattices is the normalized cross-correlation (NCC) of their respective motifs. Thus for every 2D lattice in an image, we use NCC to find the best matching 3D lattice based on motif appearance.

5. Recovering Camera Pose

For every hypothesized match between a 2D lattice detected in the image and a 3D lattice in the database, we compute a family of camera poses consistent with this match. In particular, we obtain a single rotation matrix R_w^c but a *set* of

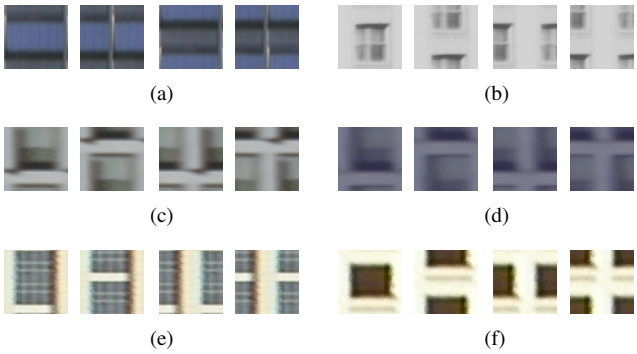

Figure 5. Database Motifs. These sets of motifs describe the appearance and capture the inherent symmetries of each building facade in the database (Figure 2). Each lattice has a unique set of motifs that enable matching between 2D and 3D lattices. Motif sets (c) and (d) represent two similar facades on the same building – an ambiguity to which our method is robust (Section 6).

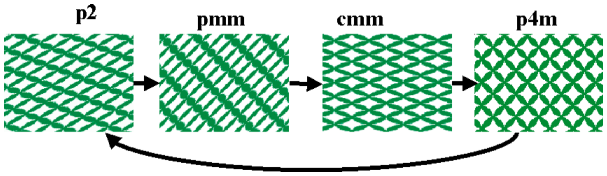

Figure 6. When a pattern is deformed by affine transformations, its symmetry group migrates to different groups within its orbit [9], in this case $p2 \rightarrow pmm \rightarrow cmm \rightarrow p4m$. The group labels are classic notations for crystallographic groups. For details see [3]. Building facades most often belong to the *pmm* and *p4m* wallpaper groups.

3D camera locations $\{T_{ci}\}$ that each correspond to a different chosen offset between the 2D and 3D lattices.

5.1. Recovering the Camera Rotation

To compute the camera rotation R_w^c we closely follow the exposition in [2]. To this end, we currently assume rectangular lattices, although not necessarily aligned with any particular world direction. From the lattice matching we obtain the images v_1 and v_2 of the vanishing points corresponding to the directions T_1 and T_2 in the world frame, respectively the image of $[1000]^T$ and $[0100]^T$ in the 3D lattice frame. We express v_1 and v_2 relative to the principal point (u_0, v_0), which can be obtained either from radial distortion pre-processing or simply assumed to be in the center of the image. We then use the derivations in [2] to obtain the rotation R_l^c *from the lattice* to the camera frame:

In case the focal length f is known, we can simply compute R_l^c by projecting the three orthogonal vanishing directions in lattice space to the camera, i.e.,

$$\begin{bmatrix} \lambda_1 x_1 & \lambda_2 x_2 & \lambda_3 x_3 \\ \lambda_1 y_1 & \lambda_2 y_2 & \lambda_3 y_3 \\ \lambda_1 & \lambda_2 & \lambda_3 \end{bmatrix} = \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix} R_l^c \qquad (1)$$

where the $[x_i\, y_i\, 1]^T \triangleq v_i$ are the (centered) vanishing points and the λ_i are the unknown projective depths. If we define

$$r_1 \triangleq \begin{bmatrix} x_1 \\ y_1 \\ f \end{bmatrix} \text{ and } r_2 \triangleq \begin{bmatrix} x_2 \\ y_2 \\ f \end{bmatrix}$$

it is easily seen starting from (1) that

$$R_l^c = \begin{bmatrix} \frac{r_1}{\|r_1\|} & \frac{r_2}{\|r_2\|} & \frac{r_1 \times r_2}{\|r_1 \times r_2\|} \end{bmatrix}$$

Note that we avoided having to know the third vanishing point v_3 by computing the third column of R_l^c as the cross product of the two first columns.

If the focal length is *unknown*, the orthogonality of the first and second column of R_l^c yields this constraint on f:

$$r_1^T r_2 = x_1 x_2 + y_1 y_2 + f^2 = 0$$

and hence f can be estimated as

$$f = \sqrt{-(x_1 x_2 + y_1 y_2)}$$

It is also worth noting that the above can be used to reject detected lattices which give imaginary values for f.

Finally, the world to camera rotation R_w^c can be computed as $R_w^c = R_l^c \left(R_l^w\right)^T$, where the lattice rotation matrix R_l^w is computed from the lattice directions T_1 and T_2 and the vector $T_3 \triangleq T_1 \times T_2$, orthogonal to the lattice plane:

$$R_l^w = \begin{bmatrix} \frac{T_1}{\|T_1\|} & \frac{T_2}{\|T_2\|} & \frac{T_3}{\|T_3\|} \end{bmatrix}$$

5.2. Recovering a Family of Translations

Once the rotation has been recovered, the set of compatible translations $\{T_{ci}\}$ can be computed from the estimated planar homography H_w^i between points on the 3D lattice and the image. Points P on the lattice plane satisfy both the lattice plane equation (with T_3 defined as above)

$$T_3^T P = d$$

and d a constant, and project to points p in the image as

$$p = K R_w^c (P - T_c)$$

Combining both, one can show (in analogy to [11] but adapted here to our conventions) that all lattice points P are mapped to image points p by a *planar homography* H_w^i:

$$p = K R_w^c (I - \frac{1}{d} T_c T_3^T) P \triangleq H_w^i P \qquad (2)$$

Once a correspondence between the 3D lattice and its 2D image is set up, the homography H_w^i is easily estimated using the conventional DLT method [5], rejecting potential outliers using RANSAC [1]. However, in doing so we can

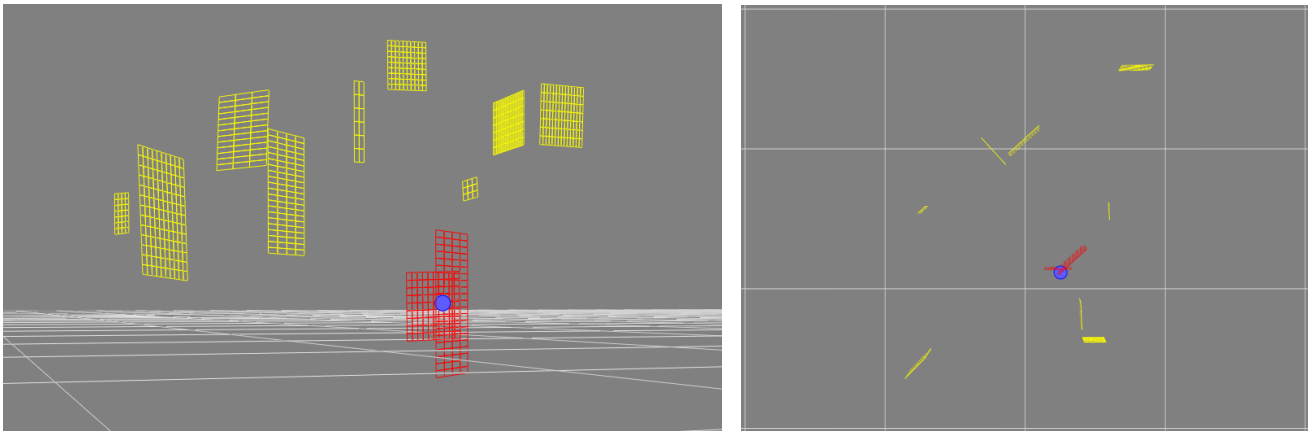

Figure 7. Families of Camera Centers. Here we see two views of a database of 3D lattices (yellow) and two families of camera locations (red) induced by two of the lattices detected in the first image of Figure 8. Note that each 2D-to-3D lattice correspondence induces a family of possible camera locations which is arranged in a pattern that reflects the symmetry of the lattice itself. The ground truth camera location (blue) should lie at the intersection of these families.

arbitrarily shift the lattice by any integer offset, i.e., we are only able to recover T_c up to an arbitrary integer translation $T_{ci} = T_c + \alpha T_1 + \beta T_2$, with α and β integers.

The canonical image location T_c corresponding to an arbitrarily chosen offset can be computed from a properly scaled version of the estimated homography H_w^i. While the DLT can only estimate H_w^i up to a scale, it can be shown [11, p. 135] that we can properly scale the result as

$$H_w^i = \frac{1}{\sigma_2(H)} H$$

with H the homography resulting from the DLT, and $\sigma_2(H)$ the second singular value of H. The latter can be computed using SVD, and the sign ambiguity is resolved by enforcing the positive depth constraint [11, p. 136]. Finally, from H_w^i we obtain the canonical camera position by solving equation 2 for camera location T_c:

$$T_c = \left[I - (R_w^c)^T K^{-1} H_w^i \right] (dT_3)$$

6. Multiple Lattices

Each correspondence between a 2D lattice in an image and a 3D lattice in the database gives us a family of potential camera locations that are themselves laid out in a lattice that reflects the geometry of the wallpaper pattern being observed (see Figure 7). When multiple building facades are visible in an image, we then have multiple families of camera locations that should intersect at a single ground position that reflects the true integer offset between each detected 2D lattice and its 3D counterpart. This point of intersection is the true camera location for the image, and thus we have geo-tagged the image.

In practice, noisy estimates of focal length prevent all families from converging at a single point. Thus, we find the camera location that minimizes the distance to the nearest camera location in each family. In addition, if all visible facades are vertical, there will be a vertical column of equally valid camera locations. However, because these locations all have the same ground position (latitude and longitude) and differ only in their height above the ground (altitude), this does not present a problem for the task considered here.

To achieve robustness in the presence of incorrect 2D-to-3D lattice correspondences, we require consistency in the camera rotation estimate induced by each correspondence. Because each 2D-to-3D lattice correspondence induces a single camera rotation R_w^c (Section 5.1), and because we have multiple 2D-to-3D lattice correspondences, we also have multiple *independent* estimates of the camera rotation. Thus, given a set of putative lattice correspondences, we select the subset with the largest support for the same camera rotation, rejecting any outliers.

In the case that only a single lattice is visible in an image, we cannot resolve the lattice offset ambiguity and determine a single camera location. However, the ground distance between the true camera location and a randomly chosen camera location from the set of possibilities will be no more than the physical width of the visible building facade. The same applies to the case that all visible lattices are parallel, thus preventing the families of camera locations from intersecting each other at a single ground position.

7. Results

We tested our method in an urban environment in which a variety of building facades are visible. The database we use consists of nine facades from seven buildings (as depicted in Figure 7) and is a subset of the textured facades depicted in Figure 1. The set of test images consisted of five 1600×1200 images (see Figures 3 and 8) for which

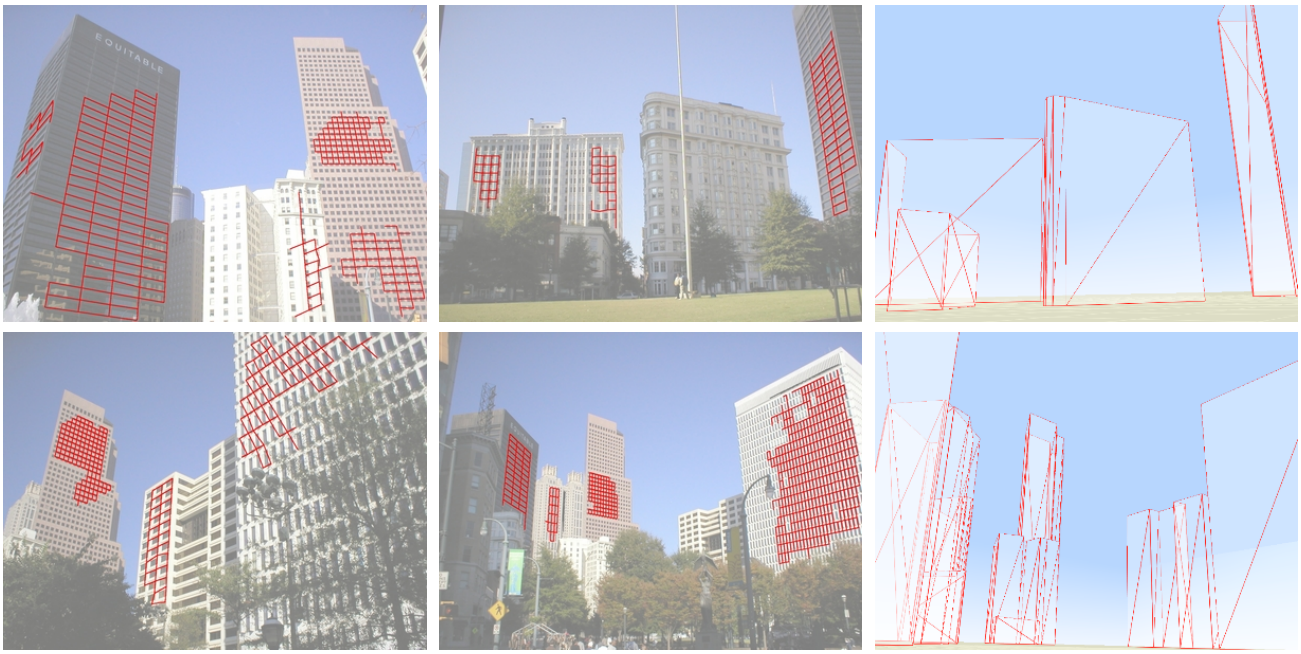

Figure 8. Four out of five images used for testing (the other is shown in Figure 3). On the right, the geo-tagging is qualitatively illustrated by rendering a synthetic image from the recovered viewpoint. Note that despite the presence of falsely detected lattices in these images, the correct 2D-to-3D lattice correspondences are discovered, enabling recovery of camera poses (see Figure 9).

we have ground truth locations. The ground truth location of each camera was determined using structure from motion with manual correspondences, and these locations are depicted in blue in Figure 9.

For the five test images, we detect 2D lattices using $N = 50$ clusters of SIFT features, match with database lattices via motifs, and recover a camera rotation and location. The mean error between ground truth and recovered camera locations was 6.04 meters, with individual errors of 2.5, 2.8, 7.3, 8.1, and 9.5 meters for the camera locations of each respective image (see Figure 9). Not only is this competitive with the errors achieved by GPS based systems, but it succeeds in precisely the types of urban environments in which GPS is susceptible to failure. In terms of camera orientation, the mean errors for compass heading and tilt were 1.51 degrees and 0.75 degrees, respectively.

Note that despite the presence of falsely-detected lattices (e.g. in feature-rich foliage) and facade-ambiguity (e.g. multiple identical facades from the same building), we are able to find the correct set of 2D-to-3D lattice correspondences by requiring rotational consistency in the matches proposed by motif matching.

8. Conclusion

We proposed a novel method to automatically geo-register photographs of man-made environments. The main contribution is the exploitation of the fundamental properties of repeating patterns throughout all stages of our algorithm – detection, matching, and pose recovery – to overcome the problems normally associated with highly repetitive environments. We have shown how to overcome the ambiguities inherent within each periodic pattern as well as between the virtually identical periodic patterns on the multiple facades of a single building.

Our method performs best when surrounded by tall buildings that exhibit highly repetitive structure, which are the exact conditions that can lead to the failure of GPS devices and traditional wide-baseline matching techniques, respectively. In addition, our method can also be applied to imagery in which GPS is not applicable, e.g., historical imagery or images posted online by amateur photographers.

In future work, we hope to extend the number of buildings and demonstrate the applicability of our method on large-scale urban databases. We also hope to include buildings from multiple cities, so that we might automatically geo-locate any photograph taken in any major urban area.

References

[1] R. Bolles and M. Fischler. A RANSAC-based approach to model fitting and its application to finding cylinders in range data. In *Intl. Joint Conf. on AI (IJCAI)*, pages 637–643, Vancouver, BC, Canada, 1981.

[2] R. Cipolla, T. Drummond, and D.P. Robertson. Camera calibration from vanishing points in images of architectural scenes. In *British Machine Vision Conf. (BMVC)*, 1999.

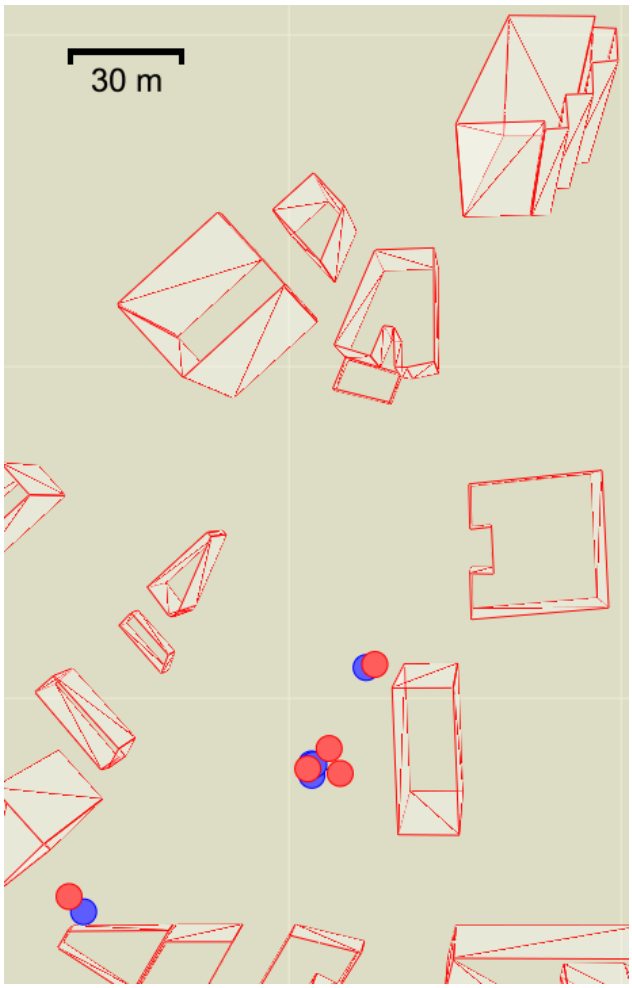

Figure 9. Recovered Camera Poses vs. Ground Truth. All recovered camera poses (red) fall within 10 meters of their ground truth positions (blue). The mean distance error was 6.04 meters. The mean errors for compass heading and tilt were 1.51 degrees and 0.75 degrees, respectively.

[3] H.S.M. Coxeter and W.O.J. Moser. *Generators and Relations for Discrete Groups*. Springer-Verlag, New York, fourth edition, 1980.

[4] T. Goedeme, T. Tuytelaars, and L. Van Gool. Fast wide baseline matching for visual navigation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 24–29, June 2004.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[6] J.H. Hays, M. Leordeanu, A.A. Efros, and Y. Liu. Discovering texture regularity as a higher-order correspondence problem. In *Eur. Conf. on Computer Vision (ECCV)*, 2006.

[7] J. Kosecka and W. Zhang. Extraction, matching, and pose recovery based on dominant rectangular structures. *CVGIP:Image Understanding*, 100(3):274–293, December 2005.

[8] Y. Liu, R. Collins, and Y. Tsin. A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Trans. Pattern Anal. Machine Intell.*, 26(3):354 – 371, March 2004.

[9] Y. Liu and R. T. Collins. Skewed symmetry groups. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 872–879, June 2001.

[10] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. J. of Computer Vision*, 60(2):91–110, 2004.

[11] Y. Ma, S. Soatto, J. Kosecka, and S.S. Sastry. *An Invitation to 3-D Vision*. Springer, 2004.

[12] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conf. (BMVC)*, pages 414–431, 2002.

[13] P. Pritchett and A. Zisserman. Matching and reconstruction from widely separated views. In *SMILE 98 European Workshop on 3D Structure from Multiple Images of Large-Scale Environments, Freiburg, Germany*, 1998.

[14] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *Intl. Conf. on Computer Vision (ICCV)*, pages 754–760, 1998.

[15] D. Robertson and R. Cipolla. An image-based system for urban navigation. In *BMVC*, 2004.

[16] F. Schaffalitzky and A. Zisserman. Geometric grouping of repeated elements within images. In *British Machine Vision Conf. (BMVC)*, pages 13–22, 1998.

[17] F. Schaffalitzky and A. Zisserman. Planar grouping for automatic detection of vanishing lines and points. *Image and Vision Computing*, 18:647–658, 2000.

[18] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *Eur. Conf. on Computer Vision (ECCV)*, pages 414–431. Springer-Verlag, 2002.

[19] T. Tuytelaars, M. Proesmans, and L. Van Gool. The cascaded hough transforms. In *ICIP*, pages 736–739, 1998.

[20] T. Tuytelaars and L. Van Gool. Wide baseline stereo based on local, affinely invariant regions. In *British Machine Vision Conf. (BMVC)*, pages 412–422, 2000.

[21] W. Zhang and J. Kosecka. Image based localization in urban environments. In *3D Data Processing Visualization and Transmission (3DPVT)*, May 2006.