Translation-Symmetry-based Perceptual Grouping with Applications to Urban Scenes

Minwoo Park†, Kyle Brocklehurst†, Robert T. Collins†, and Yanxi Liu†*

†Dept. of Computer Science and Engineering, *Dept. of Electrical Engineering
The Pennsylvania State University, University Park, PA 16802, USA
{mipark,brockleh,rcollins,yanxi}@cse.psu.edu

Abstract. An important finding in our understanding of the human vision system is perceptual grouping, the mechanism by which visual elements are organized into coherent groups. Though grouping is generally acknowledged to be a crucial component of the mid-level visual system, in computer vision there is a scarcity of mid-level cues due to computational difficulties in constructing feature detectors for such cues. We propose a novel mid-level visual feature detector where the visual elements are grouped based on the 2D translation subgroup of a wallpaper pattern. Different from previous state-of-the-art lattice detection algorithms for near-regular wallpaper patterns, our proposed method can detect multiple, semantically relevant 2D lattices in a scene simultaneously, achieving an effective translation-symmetry-based segmentation. Our experimental results on urban scenes demonstrate the use of translation-symmetry for building facade super-resolution and orientation estimation from a single view.

1 Introduction

Symmetry is an essential concept in perception and a ubiquitous phenomenon present in all forms and scales in the real world, from galaxies to atomic structures [1]. Symmetry also is considered a preattentive feature [2] that enhances object recognition. Much of our understanding of the world is based on the perception and recognition of repeated patterns that are generalized by the mathematical concept of symmetry [3].

A translation-symmetry is a translation transformation that keeps a pattern setwise invariant [4]. Mathematically, such a pattern has to be periodic and infinite. In practice, we view a finite portion of a periodic pattern in an image as an occluded infinite pattern, thus the term 'translation-symmetry' is equally applicable [5]. 2D translation symmetry detection (lattice detection) has been gaining more attention in computer vision and computer graphics in recent years [5–18]. The underlying topological lattice structure of a near-regular texture (NRT) under a set of geometric and photometric deformation fields was first acknowledged and used by Liu et al. for texture analysis and manipulation [6, 19]. Subsequently, Hays et al. [7] developed the first deformed lattice detection algorithm for real images without pre-segmentation. Hays et al. [7] formulated the lattice detection problem as a higher order correspondence problem using a spectral method that produces impressive results. Later, Park et al. [8, 9] formulated 2D deformed lattice finding as an inference problem on a Markov Random Field (MRF) and showed improved speed and accuracy on single lattice detection. Regular lattice detection has also been formulated by Han et al. [10] using statistical model selection.

In applications, Shindler et al. [15] use lattice detection to geo-tag user photos and many efforts have been made to remove clutter from real world 2D lattices and synthesize new views [14, 20]. Canada et al. [11] developed lattice detection for automatic high throughput analysis of histology array images. Liu et al. [21] apply a lattice detection algorithm to detect and remove a fence region that occludes interesting objects behind the fence.

However, state-of-the-art lattice detection algorithms cannot detect multiple lattices in the scene, which prevents wide applicability of 2D translation symmetry features for many computer vision and graphics applications. In this paper we present, for the first time, an algorithm for detecting multiple 2D lattices.

2 Translation-Symmetry-based Perceptual Grouping

The human visual system can detect many classes of patterns and statistically significant arrangements of image elements. Perceptual grouping refers to the ability to extract significant image relations and structure from lower-level primitive image features without prior knowledge of high-level image content. Our proposed method follows this concept. We first detect lower-level primitive image features such as Kanade Lucas Tomasi corners (KLT) [22], Maximally Stable Extremal Regions (MSER) [23], and Speeded Up Robust Features (SURF) [24]. Then, each set of feature points is grouped by that feature's descriptor and 2D lattice structures are proposed from each group. The proposed grouping method is an iterative procedure similar to a standard clustering algorithm such as K-means or mean-shift clustering, except that the similarity metric reflects higher-level knowledge of 2D translation symmetry such as texel appearance, (t_1, t_2) basis vector pair, and lattice coverage in the image. Once we obtain this

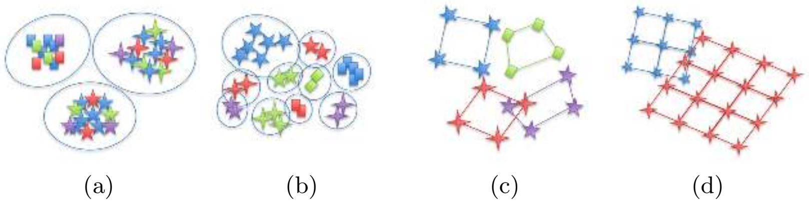
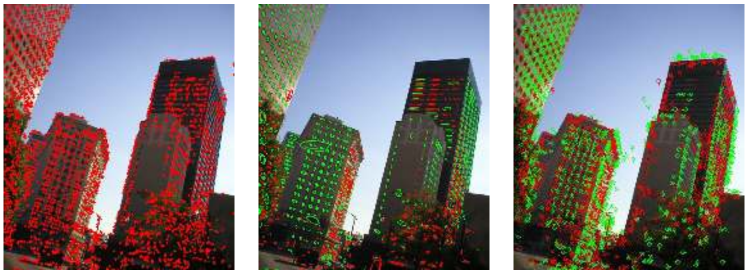

(a) (b) (c) (d)

Fig. 1: (a) Lower-level visual primitives (KLT, MSER, and SURF) (b) Visual grouping of each type of feature (c) (t_1, t_2) basis grouping by RANSAC (d) 2D lattice completion and grouping.

information, we can rectify the perspective distortion of the 2D translation symmetry as well as collect additional valid lattice points that were not detected by any of the low-level features detectors. This increases the quality of the detected 2D translation symmetry.

2.1 Low-level Feature Aggregation

The use of different types of lower-level primitive features is beneficial because KLT features, MSER, MSER on the inverted image, and SURF with a positive or negative laplacian generate different responses to different visual elements, and therefore we can reliably find a wide range of 2D lattice points. As can be seen in Figure 2, some of the valid 2D lattice points are only identified by one or two of the detector types, thus justifying using a set of complementary feature detectors.



(a) KLT (b) MSER (c) SURF

Fig. 2: Low-level primitive visual features detected by KLT, MSER, and SURF. Some of the valid lattice points are not identified by all of the feature detectors but only a subset of them. MSER and MSER on the inverted image are displayed in green and red, respectively, and SURF features with a positive and negative laplacian are colored red and green.

2.2 Grouping of Low-level Features

Since the number of repeating patterns is not given a priori, we use the mean-shift algorithm with a varying bandwidth to cluster the different types of lower-level features. Since KLT only specifies the 2D location of points and MSER only gives a 2 by 2 scatter matrix of the region, we extract 11 by 11 subimages centered at each KLT feature and the center of the MSER region. Each subimage is normalized by subtracting the mean pixel value, and dividing by the standard deviation of pixel values to compensate for illumination changes.

2.3 Translation-Symmetry-based Grouping

We seek a (t_1, t_2)-vector pair that represents the generators of the translation symmetry subgroup using a RANSAC-based method, similar to the work of Park

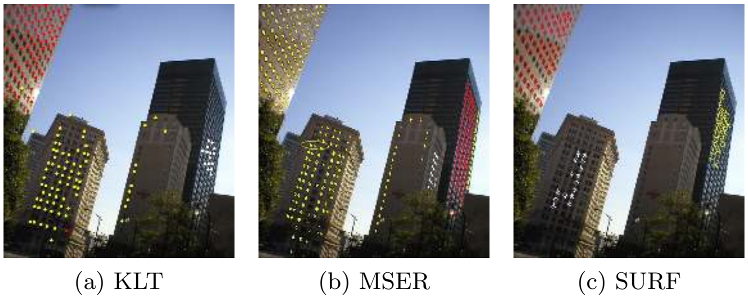


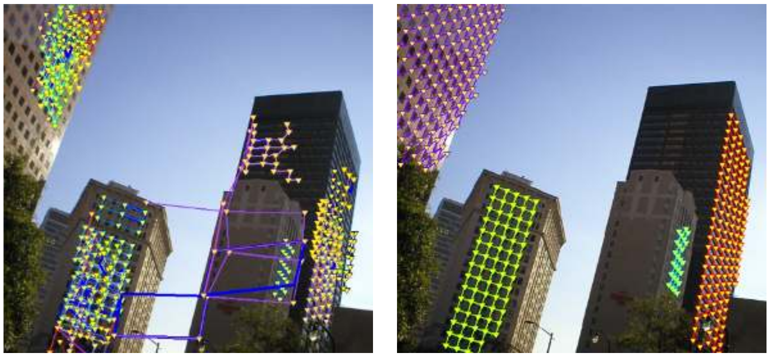
Fig. 3: Sample results of mean-shift clustering of low-level features. For clarity we manually choose clusters that are on the 2D lattice structures.

et al. [9] and Schindler et al. [15]. Schindler et al. [15] randomly select 4 points from a set of SIFT features, whereas Park et al. [9] improve this random proposal by considering proximity of KLT points to avoid proposals with an invalid affine transformation. We further examine whether the proposed 4 points form a valid quadrilateral to increase the likelihood of finding a feasible perspective mapping. Using this proposal, we iteratively complete the 2D lattice structure under a perspective deformation model while allowing some tolerance using a normalized threshold that is independent of the image.

Proposals of a basis quadrilateral: For each detected feature point cluster, we randomly sample three points $\{a; b; c\}$ to form a $(t_1,\, t_2)$ vector pair given by $b-a$ and $c-a$, compute the fourth point, d given by t_1+t_2+a, and compute the perspective transformation that maps these four points from image space into the integer lattice basis $\{(0,0),(1,0),(0,1),(1,1)\}$. We can now transform all remaining points from image space into their equivalent lattice positions via the same perspective transform, and count as inlier points those whose lattice space coordinates are within some threshold[1] of an integer position (x, y) . If the four chosen points $\{a; b; c; d\}$ define a valid basis quadrilateral of a 2D translational pattern, many additional supporting votes should emerge from other interest points having a similar spatial configuration.

Lattice completion: Since many of the valid lattice points are not detected by any of the lower-level primitives, we further seek to recover all missed lattice points that are not initially identified by the feature detectors. For this task we evaluate normalized cross correlation between the basis quadrilateral and input image. Note that this is not possible without the hypothesized perceptual grouping of low-level features since otherwise we do not know whether there are repeating patterns, how many there are, and what they look like. Due to possible foreshortening effects, identifying all of the valid lattice points in one

[1] 0.2 is used for all our experiments and this threshold is image independent since all the points are transformed to normalized coordinates (integer coordinates)



(a) Proposals of quadrilateral

(b) Final grouping

Fig. 4: Sample results of quadrilateral proposals and final grouping result: (a) Note that there are many duplicate basis quadrilaterals found on the same building. (b) These are further clustered and filtered by the proposed algorithm.

iteration using cross correlation suffers from inaccurate localization of the likelihood peaks. To avoid this problem we first rectify the image using the mapping from the current observed lattice points, $\{p_c(j,i)|1 \leq j \leq M, 1 \leq i \leq N\}$ to the regular lattice constructed by the found (t_1, t_2) basis vector pairs. The (t_1, t_2) basis vector pair and regular lattice point, $p_r(j,i)$ are given by

$$\begin{aligned} t_1 = p_c(1,2) - p(1,1), \quad t_2 = p_c(2,1) - p_c(1,1) \\ p_r(j,i) = p_c(1,1) + t_1(i-1) + t_2(j-1) \end{aligned} \tag{1}$$

We then compute a perspective mapping, H_{cr} from $p_c(j,i)$ to $p_r(j,i)$ and warp input image I_i to get a rectified image $I_r = H(I_i)$. Next, we compute a median quadrilateral, T_m from all the quadrilaterals centered at lattice coordinates $p_r(j,i)$ defined by (t_1, t_2) basis pairs. We compute the normalized cross correlation between median texel T_m and the rectified image I_r $(NCC(I_r, T_m))$ and get local peaks (x, y) by non-maxima suppression.

At this stage, the procedure becomes iterative. We propose a refined mapping, $H_{ri}^{(t)}$ from $p_r^{(t)}(j,i)$ to (j,i) at each iteration t. Only the peaks that are transformed to neighborhoods[2] of integer positions $(\hat{x}, \hat{y})$ are chosen as valid lattice points and used to update the lattice point set $p_r^{(t+1)}(j,i) = p_r^{(t)}(j,i) \cup (\hat{x}, \hat{y})$. We then recompute the rectification mapping $H_{ri}^{(t+1)}$ using correspondences between $p_r^{(t+1)}(j,i)$ and (j,i) and repeat the entire procedure until $p_r^{(t+1)}(j,i) = p_r^{(t)}(j,i)$. This is summarized by pseudo code in Figure 5.

Perceptual grouping of lattices: From all candidate proposals, $\{Pr_i|i = 1 \sim N\}$ we sort all the proposals by the normalized A-score introduced in [6]. The more that quadrilaterals in the lattice look alike and the higher the number of quadrilaterals in the lattice, the smaller the A-score. Starting from the best

[2] The same tolerance threshold as section 2.3 is used in all of our experiments.

```
set t=0, Compute $(t_1,\ t_2)$ and $p_r^{(t)}(j,i)$ by equation (1)
Compute mapping $H_{cr}$ using correspondences $p(j,i)$ to $p_r(j,i)$
Rectify the input image $I_i$ by $H_{cr}$ to get the rectified image, $I_r$
Compute median quadrilateral $T_m$
Compute normalized cross correlation $NCC(I_r, T_m)$ between $T_m$ and $I_r$.
Compute non-maximum suppressed peaks $(x, y)$ from $NCC(I_r, T_m)$
do
    compute $H_{ri}^{(t)}$ from $p_r^{(t)}(j,i)$ to $(j,i)$
    if distance between $H_{ri}^{(t)}$[x;y] and round($H_{ri}^{(t)}$[x;y]) $\leq 0.2$
        $p_r^{(t+1)}(j,i) = p_r^{(t)}(j,i) \cup (x,y)$
    end
    t=t+1
while $p_r^{(t+1)}(j,i) \neq p_r^{(t)}(j,i)$
```

Fig. 5: Pseudo code for the lattice-completion algorithm

proposal in terms of the normalized A-score, we group Pr_i while performing the lattice-completion algorithm (section 2.3). As can be seen in Figure 7, the output of the lattice-completion algorithm (section 2.3) gives rough segmentations of the scene, therefore, we use this information to group the Pr_i. Let the input lattice proposal and output lattice be Pr_i and L_i respectively and let the lattice-completion algorithm (section 2.3) be $F()$, then $L_i = F(Pr_i)$. The initial cluster center, which is a completed 2D lattice, is initialized by $L_1 = F(Pr_1)$ and we then group $\{Pr_i | 2 \leq i \leq N\}$ only when more than 70% of the 2D lattice points in Pr_i are contained in the quadrilaterals in L_1. From the Pr_i that are not grouped to the first cluster center we choose the best proposal in terms of its normalized A-score and we generate a second cluster center using the lattice-completion algorithm (section 2.3). This procedure repeats until no more ungrouped proposals are left. For example, Figure 11b has 72 proposals and the proposed method is successful in grouping all of the proposals. Pseudo code for grouping is given in Figure 6.

2.4 Quantitative Evaluation

We have compared the proposed perceptual grouping algorithm, which we will refer to as **PG**, against Park et al. [9], which we will refer to as **PAMI09**. We have tested the **PAMI09** and **PG** algorithms on a publicly available dataset containing 120 real-world urban scene images with ground-truth [9]. We evaluate the precision and recall rate of the detected lattices using the automated evaluator described in [9]. The number of true positives (TP) is given by the number of correctly identified texels, the number of false positives (FP) is given by the number of falsely detected texels, and the number of false negatives (FN) is given by the number of ground-truth texels minus the number of true positives. When N is the number of 2D lattices in the entire data set, the precision

```
For every $Pr_i$ for $1 \leq i \leq N$
Sort $Pr_i$ by normalized A-score
Enqueue each $Pr_i$ to $Q_p$, Enqueue($Q_p, Pr_i$)
Initialize queue, $Q_L$ for lattice grouping. $Q_L = NULL$
while $Q_p! = NULL$
    L=F(Dequeue($Q_p$))
    Enqueue($Q_L$,L)
    Initialize temporary queue $Q_t = NULL$
    while $Q_p! = NULL$
        P=Dequeue($Q_p$)
        if L $\supset P$
            Group P to L
        else
            Enqueue($Q_t$,P)
        end
    end
    $Q_p = Q_t$
end
```

Fig. 6: Pseudo code for perceptual grouping

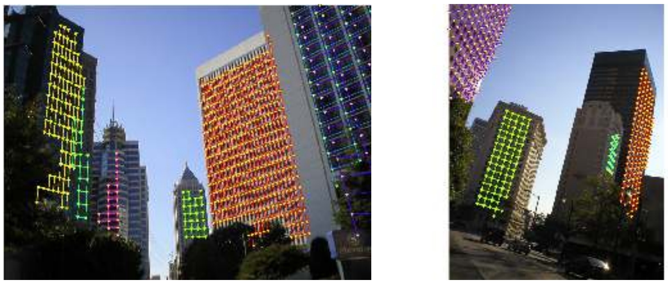
and recall rates are given as

$$Precision = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N}(TP_i + FP_i)}, \quad Recall = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N}(TP_i + FN_i)} \tag{2}$$

Instead of computing average precision and recall rates, equation (2) is used to reflect the difference between the successful detection of lattices with, for example, 1000 texels versus 4 texels.

Accuracy: We measure the detection rate of **PAMI09** and **PG** only when these two algorithms detect the same lattice structure, since **PAMI09** [9] is intended for detecting only a single deformed lattice (14672 ground-truth texels). Second, we measure detection rates against all of the ground-truth to show the multiple lattice detection capability of **PG** (23753 number of ground-truth texels). Since **PAMI09** [9] is not intended for multiple lattice detection, we first run **PAMI09** [9], then remove the portion of image where the 2D lattice is found, and repeat until no more lattices are found.

As can be seen in Figure 8, the precision rate of **PG** has improved by 8.4 % over **PAMI09** [9] for both single and multiple lattice detection and the recall rate of **PG** is improved by 10% and 20% over **PAMI09** [9] for single and multiple lattice detection respectively. In addition, the precision and recall rates of **PG** and **PAMI09** [9] for detecting multiple lattices does not drop significantly from the rates on single lattices, as can be seen in Figure 8. This effectiveness of our method comes from: 1) feature aggregation from a variety of interest point detectors, which is more reliable at exposing repeating structures; 2) modeling the deformation of the lattice by perspective projection rather than non-rigid

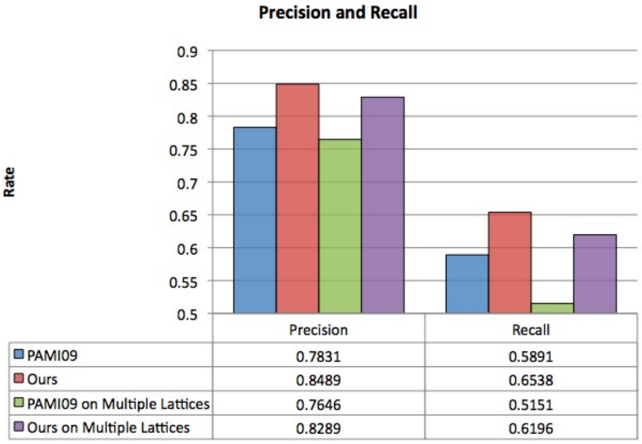


(a) (b)

Fig. 7: Sample results of translation-symmetry-based perceptual grouping are shown. Different colors mean different groups.

deformation for fast, simple, and accurate application to rigid objects; and 3) perceptual grouping of multiple lattices.

Efficiency: The **PG** algorithm takes 4.2 ± 2.07 min using a 2.4 GHz Intel P8600 4GB machine in MATLAB while **PAMI09** [9] takes 15.8 ± 11.3 min. This confirms that the new method is more efficient and more accurate.



	Precision	Recall
🟦 PAMI09	0.7831	0.5891
🟥 Ours	0.8489	0.6538
🟩 PAMI09 on Multiple Lattices	0.7646	0.5151
🟪 Ours on Multiple Lattices	0.8289	0.6196

Fig. 8: The blue and red bars indicate precision and recall rate of single lattice detection (14672 ground-truth texels) for the **PG** (red) and the **PAMI09** [9] (blue) algorithms. The green bar indicates precision and recall of sequential runs of **PAMI09** [9] and the purple bar indicates precision and recall rate of **PG** for detecting multiple lattices within a single image (23753 ground-truth texels).

3 Application

To demonstrate a possible application using the 2D lattice grouping proposed in this paper, we have used the detected lattices for single view super-resolution and urban scene analysis.

3.1 Super-Resolution from a Single View

Recently Glasner et al. [25] showed the power of super-resolution from a single view. Recurrence of similar patches in an image forms the basis for their single view super-resolution approach in [25], and therefore correctly identifying corresponding patches is very important in this. As can be seen in our results in Figure 7, we solve this correspondence problem for texels in a lattice structure.

Instead of running a state-of-the-art super-resolution algorithm such as [25], we took a basic approach where multiple images of the same scene are registered, a median image is computed, and de-blurring is performed. In our case we rectify each quadrilateral in the 2D lattice into the same coordinate system, compute a median texel, and perform deconvolution to get a high resolution (HR) image. We map each recovered HR image back to the original space and combine the existing original low resolution (LR) image to transfer high frequency HR information while retaining original lighting and shadow changes. We first perform a discrete cosine transform (DCT) on the HR image to isolate the high frequency components by truncating absolute DCT coefficients larger than 80% of the largest absolute DCT components to get truncated DCT block D[3]. Inverse DCT is then performed to get HR_{hf} and HR_{hf} is added back to the original LR image, thus preserving local information[4]. Sample results are shown in Figure 9.

3.2 Frontal View Facades Estimation from a Single View

Before we attempt to analyze an urban scene, we need to resolve ambiguity of (t_1, t_2) vector pairs under perspective distortion since there could be many choices of valid (t_1, t_2) vector pairs for a given perspective distortion of a 2D wallpaper pattern. This can make estimation of frontal facets of buildings ambiguous. We want the (t_1, t_2) vector pair to be aligned to vertical and horizontal edges of the building, since these edges are typically aligned with meaningful directions, either parallel to or perpendicular with the ground. Figure 11 (a) shows (t_1, t_2) vector pairs that are not aligned with the horizontal and vertical edges of the building. Figure 11 (b) shows (t_1', t_2') vector pairs after the desired correction. In the following section we will explain in detail how we correct (t_1, t_2) vector pairs.

[3] This is the inverse of JPEG procedure where one wants to discard high frequency information to achieve compression.

[4] For further details, please refer to our supplemental material.

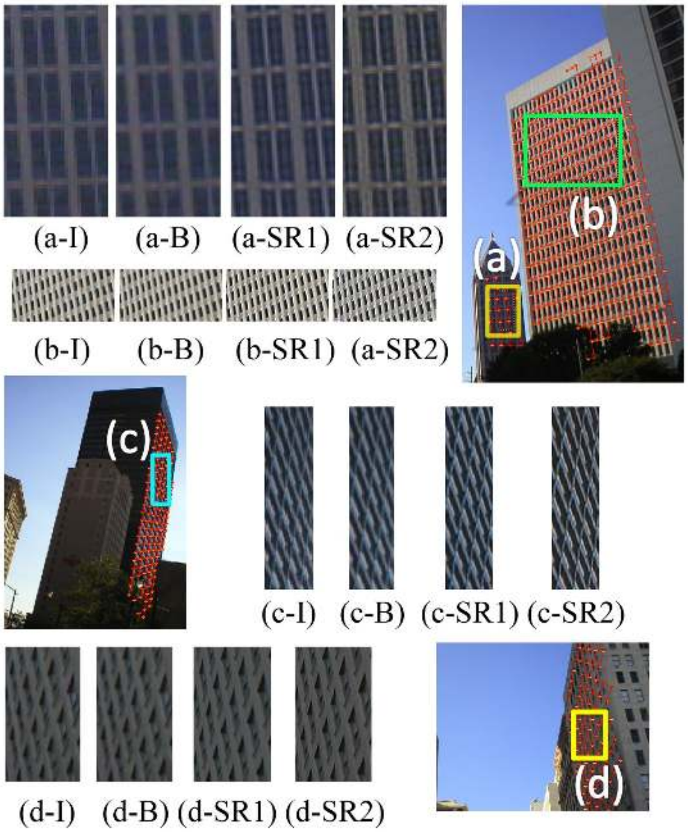


Fig. 9: Sample single view SR results are shown. I stands for original input, B stands for bicubic interpolation, SR-1 stands for super-resolution with the exact copy of the texels, SR-2 stands for super-resolution with a local information transfer such as lighting and shadow. (a-d) input selection. (a-B ∼ d-B) results of 2× bicubic interpolation. (a-SR1,2 ∼ d-SR1,2) results of 2× SR.

Resolving ambiguity of (t_1, t_2) vector pairs Most modern architecture falls into either the pmm or p4m subgroup of the 17 possible 2D wallpaper patterns [15]. In such cases, the (t_1, t_2) vector pair should be aligned with both the reflection axes and the horizontal and vertical edges of the building[5]. First, we enumerate variations of (t_1, t_2) from the current detected lattice. Let a lattice point at row j and column i be given by $p(j, i)$, then the current t_1 and t_2 are given as $t_1 = p(j, i+1) - p(j, i)$ and $t_2 = p(j+1, i) - p(j, i)$ respectively. The variation of (t_1, t_2) can be given as $t'_1 = p(j, i+1) - p(j, i)$ and $t'_2 = p(j+1, i+1) - p(j, i)$, or $t'_1 = p(j, i+1) - p(j, i)$ and $t'_2 = p(j+1, i-1) - p(j, i)$ as can be seen in Figure 10b or Figure 10c.

[5] We do not examine horizontal and vertical gradient information to correct (t_1, t_2) as there might be severe perspective distortion.

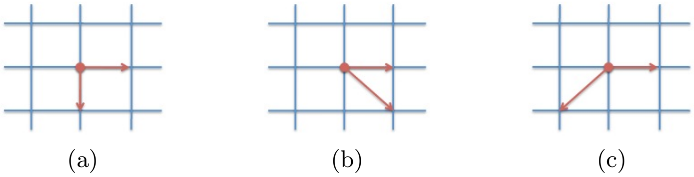
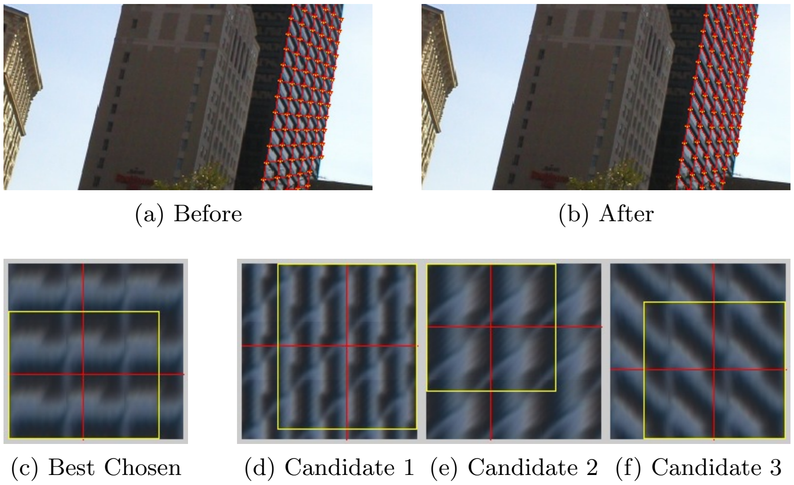


Fig. 10: (a) The current (t_1, t_2) vector given as $t_1 = p(j, i+1) - p(j, i)$ and $t_2 = p(j+1, i) - p(j, i)$ (b) The variation of (t_1, t_2) vector given as $t'_1 = p(j, i+1) - p(j, i)$ and $t'_2 = p(j+1, i+1) - p(j, i)$ (c) The variation of (t_1, t_2) vector given as $t'_1 = p(j, i+1) - p(j, i)$ and $t'_2 = p(j+1, i-1) - p(j, i)$

We then compute a median texel from quadrilaterals which have been transformed from their 4 observed points in the lattice, $\{p(j, i), p(j, i) + t_1, p(j, i) + t_1 + t_2, p(j, i) + t_2\}$, to rectified points, $\{(1, 1), (w, 1), (w, h), (1, h)\}$ where h and w are the height and width of rectified texels (both set as 50 pixels). Then we tile nine copies of the computed median texel in a 3 by 3 grid to form a small regular lattice pattern and attempt to find the two reflection axes. We only search through x and y directions near the center of the rectified median texel. This is sufficient and necessary because, if reflection axes exist, they must be parallel to the (t_1, t_2) vector pair.

We repeat this procedure for all the enumerated (t_1, t_2) vector pairs and seek reflection axes. The sum of the absolute difference between the median texel and the flipped median texel is computed and we select the (t_1, t_2) vector pair that generates the minimum sum as the best pair. Sample results are shown in Figure 11 (c,d,e,f). As can be seen in Figure 11, the analysis is successful in aligning (t_1, t_2) to the vertical and horizontal edges of the building facade.



(a) Before
(b) After
(c) Best Chosen
(d) Candidate 1
(e) Candidate 2
(f) Candidate 3

Fig. 11: Sample results of correction of (t_1, t_2) vector pair using reflection axes analysis. (a) before (b) after (c) best texel shape (d-f) candidate texel shapes

Computation of Frontal View Collins and Beveridge [26] showed that when the vanishing points of a 3D plane projected onto an image and the angular field of view of the camera are known, a 3D rotation matrix can be used to relate observed image locations on the plane to the image coordinates they would have if the plane were rotated to face the camera (having a normal vector pointing directly along the camera view direction). Their formulation shows that if the vanishing line of the plane is given by the formula $ax+by+c=0$, then the normal to that plane, in camera coordinates, is $n=(a,b,c)/\|(a,b,c)\|$. The matrix that will perform the projective transform simulating the desired 3D rotation is, in homogeneous coordinates,

$$k_i \begin{bmatrix} x'_i \\ y'_i \\ f \end{bmatrix} = \begin{bmatrix} E & F & a \\ F & G & b \\ -a & -b & c \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ f \end{bmatrix} \tag{3}$$

where

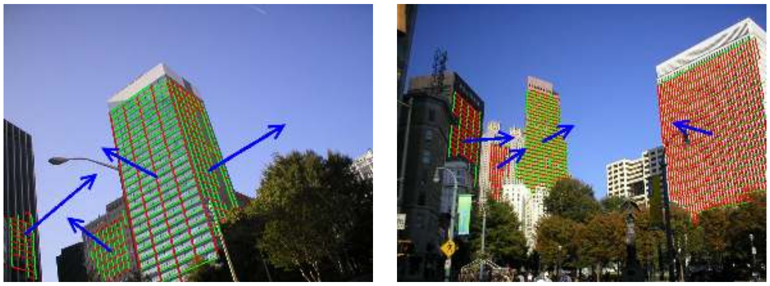
$$E = \frac{a^2c+b^2}{a^2+b^2}, F = \frac{ab(c-1)}{a^2+b^2}, G = \frac{a^2+b^2c}{a^2+b^2} \tag{4}$$

where f is the focal length given by $f = \frac{w}{2tan(FOV/2)}$ where w is the image width and FOV is the camera's angular field of view.

A perspectively distorted lattice that has been identified by our method will converge to two vanishing points, one in each direction of 2D repetition. We can calculate the vanishing points for a lattice covering the facade of a building, then calculate the line connecting the vanishing points in the form $ax+by+c=0$. From that equation, the values (a,b,c) are the normal vector to the plane in camera coordinates [26]. We use these values to draw the normal vectors to building facades in Fig. 12.

We cannot perform the projective transform that would simulate bringing the building facade into a frontal view without knowing the angular field of view of the camera. However, we assume that the two directions of repetition on a building facade are orthogonal in a frontal view. If an incorrect field of view were assumed and used to bring the lattice into a frontal view, the two directions generating the lattice would not be orthogonal. Specifically, an incorrect field of view used to generate the frontal view will induce a scaling along the direction of the facade normal in image coordinates. In our supplemental material, we show that a simple search routine can quickly converge upon the one unique value for angular field of view that can be used to bring a lattice into a frontal view while preserving the orthogonality of the directions of 2D repetition. We show the computed size and shape of the lattice and texels for three images in Fig. 12.

This is a powerful application of our method because computation of building facade normals can be used for 3D reconstruction and geotagging. The calculation of a frontal view of a building facade also can enable extraction of the building appearance as a 2D texture, and can be useful for building recognition where only the frontal appearance of a building is known.



(a) (b)

Fig. 12: This figure shows the computation of surface normals from a lattice detected on a building facade. The blue arrow indicates the surface normal of the building.

4 Conclusion

A novel 2D translation-symmetry-based method of perceptual grouping is presented that shows superior performance in terms of detecting single and multiple lattices in an image over the state-of-the-art algorithm. Perceptual grouping is possible when mid-level information of scene structures is successfully obtained. Also, we have demonstrated that the detected lattice structure can be used for single view super-resolution as well as for 3D orientation estimation in urban scenes. We plan to extend this work on single view 3D urban scene reconstruction and apply mid-level visual features for object categorization.

Acknowledgement This work is supported in part by an NSF grant IIS-0729363 and a Google Research Award to Dr. Liu.

References

1. Gardner, M.: The new ambidextrous universe: symmetry and asymmetry, from Mirrow reflections to superstrings. W.H. Freeman and Company (1979)
2. Conners, R., Ng., C.: Developing a quantitative model of human preattentive vision. Volume 19. SMC (1989)
3. Weyl, H.: Symmetry. Princeton University Press, Princeton (1952)
4. Grunbaum, B., Shephard, G.: Tilings and Patterns. New York: W.H. Freeman and Company (1987)
5. Liu, Y., Hel-Or, H., Kaplan, C.S., Van Gool, L.: Computational symmetry in computer vision and computer graphics. Foundations and Trends in Computer Graphics and Vision **5** (2010) 1–156
6. Liu, Y., Collins, R.T., Tsin, Y.: A computational model for periodic pattern perception based on frieze and wallpaper groups. Pattern Analysis and Machine Intelligence, IEEE Transactions on **26** (2004) 354–371
7. Hays, J., Leordeanu, M., Efros, A., Liu, Y.: Discovering texture regularity as a higher-order correspondence problem. In: 9th European Conference on Computer Vision. (2006) 522–535

8. Park, M., Collins, R.T., Liu, Y.: Deformed Lattice Discovery via Efficient Mean-Shift Belief Propagation. In: European Conference on Computer Vision, Marsellie, France (2008)
9. Park, M., Brocklehurst, K., Collins, R., Liu, Y.: Deformed Lattice Detection in Real-World Images Using Mean-Shift Belief Propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence **31** (2009) 1804–1816
10. Han, J., McKenna, S., Wang, R.: Regular texture analysis as statistical model selection. In: 10th European Conference on Computer Vision, Marsellie, France (2008)
11. B.A. Canada, G.K. Thomas, K.C.J.W., Liu, Y.: Automatic lattice detection in near-regular histology array images. In: Proceedings of the IEEE International Conference on Image Processing. (2008)
12. Mitra, N.J., Guibas, L., Pauly, M.: Partial and approximate symmetry detection for 3d geometry. In: ACM Transactions on Graphics. Volume 25. (2006) 560–568
13. Schaffalitzky, F., Zisserman, A.: Geometric grouping of repeated elements within images. In: Shape, Contour and Grouping in Computer Vision. (1999) 165–181
14. Korah, T., Rasmussen, C.: Analysis of building textures for reconstructing partially occluded facades. In: ECCV '08: Proceedings of the 10th European Conference on Computer Vision, Berlin, Heidelberg, Springer-Verlag (2008) 359–372
15. Schindler, G., Krishnamurthy, P., Lublinerman, R., Liu, Y., Dellaert, F.: Detecting and Matching Repeated Patterns for Automatic Geo-tagging in Urban Environments. In: Computer Vision and Pattern Recognition. (2008) 1–8
16. Leonard G. O., H., Kanade, T.: Computer analysis of regular repetitive textures. In: Proceedings of a workshop on Image understanding workshop, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1989)
17. Lin, H.C., Wang, L.L., Yang, S.N.: Extracting periodicity of a regular texture based on autocorrelation functions. In: Pattern Recognition Letters. (1997) 433–443
18. Leung, T., Malik, J.: Detecting, localizing and grouping repeated scene elements from an image. In: 4th European Conference on Computer Vision. (1996) 546–555
19. Liu, Y., Tsin, Y., Lin, W.C.: The Promise and Perils of Near-Regular Texture. International Journal of Computer Vision **62** (2005) 145 – 159
20. Tsin, Y., Liu, Y., Ramesh, V.: Texture replacement in real images. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 2. (2001) 539 – 544
21. Liu, Y., Belkina, T., Hays, J., Lublinerman, R.: Image De-fencing. (In: IEEE Conference on Computer Vision and Pattern Recognition, 2008.) 1–8
22. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition. (1994) 593–600
23. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC. (2002)
24. Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. In: In ECCV. (2006) 404–417
25. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: ICCV. (2009)
26. Collins, R.T., Beveridge, J.R.: Matching perspective views of coplanar structures using projective unwarping and similarity matching. In: In Proc.Int.Conf. of Computer Vision and Pattern Recognition, CVPR. (1994) 240–245