Image De-fencing Revisited

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. We introduce a novel image defencing method suitable for consumer photography, where plausible results must be achieved under common camera settings. First, detection of lattices with see-through texels is performed in an iterative process using online learning and classification from intermediate results to aid subsequent detection. Then, segmentation of the foreground is performed using accumulated statistics from all lattice points. Next, multi-view inpainting is performed to fill in occluded areas with information from shifted views where parts of the occluded regions may be visible. For regions occluded in all views, we use novel symmetry-augmented inpainting, which combines traditional texture synthesis with an increased pool of candidate patches found by simulating bilateral symmetry patterns from the source image. The results show the effectiveness of our proposed method.

1 Introduction

We address a real-life problem in photo editing where one would like to remove or change fence-like, near-regular foreground patterns that are often unavoidable, as illustrated in Figure 1. This task was first addressed by Liu et al. [1] by a 3-step procedure 1) lattice detection [2], 2) foreground / background segmentation and 3) inpainting [3, 4]. Lattice detection and foreground/background segmentation in [1] proceeded sequentially, hence an abundant amount of information arising from the repeating pattern was not fully utilized. Furthermore, the performance of previous lattice detection algorithms [2] is far from practical for this application due to inaccuracy and slowness.

In this paper, we make the following novel contributions to this challenging goal; • **online learning and classification** is used to aid lattice detection and segmentation, resulting in a substantial improvement in detection rate over current state-of-the-art lattice detection algorithms [5, 2]. Our online classification and segmentation method is not confined to this specific application; it can be applied to other near-regular texture detection and analysis tasks. • **multiview inpainting** is introduced to improve the region filling process by using multiple, shifted camera views, since the best way to infer an unknown pixel is to see the occluded region in another view. The approach does not assume any rigidity of the fence nor objects, but requires some offset between views: either by camera or object movement. These are practical requirements for every-day photography, since one can take multiple photos of a scene simply by shifting the camera,

(a) Input (b) Automatic Segmentation

(c) Liu et al. [1] (d) Our Result

Fig. 1: (a) Input image (b) Automatic segmentation using online learning (c) Result of Liu et al. [1] (d) Result of our proposed method

revealing objects behind the fence due to parallax. • **symmetry-augmented inpainting** is introduced to tackle the problem of scarcity of candidate samples after large amounts of foreground have been removed leaving fragmented background pixels. We increase the candidate pool by simulating bilaterally symmetric patches from the source image. For instance, if half of someone's mouth is covered, we can recover the occluded region reliably from the opposite side of the mouth by reflecting that patch. The experimental results show the effectiveness of our proposed method, especially for objects that are extremely unforgiving to flawed inpainting such as a human face and structured backgrounds (see Figures 1 and 8 for examples).

2 Related Work

Liu et al. [1] introduce a novel application in computational photography by taking advantage of see-through NRTs to remove a near regular foreground. As the authors of [1] point out, each of the components in the application is very challenging on its own and poses many research questions.

2.1 Lattice Detection

There is a rich body of work on lattice detection in the literature [6, 2, 5, 7–11]. However, it was Hays et al. [2] who first developed an automatic deformed lattice

detection algorithm for real images without pre-segmentation. The method of [2] is based on looking for the neighbors of a randomly selected interest point in the image. If a sufficient number of points look like their respective t_1, t_2 neighbors (lower order similarity) and also share their t_1, t_2 neighbors' directions/orientations (higher order correspondences) towards other interest points in the image, those points and their neighborhood relationships are confirmed to be part of the lattice. Based on this partial result, the slightly deformed lattice is straightened out and a new round of lattice discovery starts, so the extracted lattice grows bigger and bigger. Formulating the lattice detection problem as a higher order correspondence problem adds computational robustness against geometric distortions and photometric artifacts in real images, and the publicly available code produces impressive results.

Later, Park et al. [5] developed a deformed lattice detector within a Markov Random Field using an efficient inference engine called Mean-Shift Belief Propagation. They showed 72% improvement in lattice detection rate over the Hays' algorithm [2], with a factor of 10 speed up.

However, all algorithms discussed so far ignore the foreground/background characteristics of the repeating pattern we want to find. In particular, images which contain fence-like structures are inevitably highly irregular despite the regularity of the foreground. For such cases, the irregular background interferes with the detection of the see-through foreground lattice. Our method learns the type of the repeating pattern, removes the irregularities, and uses the learned regularity in evaluating the foreground appearance likelihood during lattice growth, a crucial improvement since robust and complete lattice detection plays the most significant role in our application.

2.2 Image Completion

Traditional texture filling tools such as Criminisi et al. [3, 4] require users to manually mask out unwanted image regions. Based on our own experience, for images such as those in Figures 1, 7, and 8, this process is very tedious and error-prone. Simple color-based segmentations are not sufficient. Painting a mask manually, as in previous inpainting work, requires copious time and attention because of the complex topology of the foreground regions.

Favaro et al. [12] introduce a method for the restoration of images in which certain areas have been blurred. Their method develops a map of the relative amount of blur at each position in the image, then learns correspondences between recurring objects or image patches. This allows them to copy the least blurred occurrence of an object and paste patches from it to inpaint over blurred occurrences of the same or similar objects. This is a powerful method of relating undesirable blur utilizing the power of understanding multiple instances of the same object in a scene. Their work differs from ours in that they do not attempt to use or understand any underlying structure, such as a lattice, that may exist among the instances of the recurring object. Also, their method of inpainting removes blurring, such as that from varying depth, but does not remove occlusion, such as a fence-like foreground region.

As an extension to photo inpainting, Wexler et al. [13] and Patwardhan et al. [14] each propose a video inpainting method. This is desirable, since temporal information can give additional information that can aid the inpainting process. Although the balance of spatial and temporal continuity is far from trivial, both methods produced spatially and temporally coherent results, albeit at the cost of needing to mask out unwanted regions manually. With these filling tools, a user has the capability to reveal content in a photo behind occlusions. However, if the missing region is part of a complex object with high resolution, such as a human subject, the quality of inpainting is often insufficient, as can be seen in Figure 1 and 8.

Hays and Efros [15] proposed a scene completion method using millions of photographs. The algorithm fills in the hole regions in images with seamless and semantically valid patches from the database. However neither the database images nor the regions to be filled are fragmented by any foreground structures.

Vaish et al. [16] proposed a method to reconstruct densely occluded scenes using synthetic aperture photography. However, they require a large, synchronized camera array ($30 \sim 100$ cameras) to achieve this goal, which is obviously impractical for consumer-grade use.

Our approach represents a middle ground between traditional image completion and video completion/synthetic aperture reconstruction, since we use only a small number of auxiliary images that are easily achievable in everyday photography.

3 Near Regular Texture Segmentation

Our basic lattice detection algorithm is similar to [5]. The procedure is divided into two phases, where the first phase proposes one (t_1 , t_2)-vector pair and one texture element, or texel. 2D lattice theory tells us that every 2D repeating pattern can then be reconstructed by translating this texel along the t_1 and t_2 directions. During phase one, we detect KLT corner features, extract texture around the detected corners, and select the largest group of similar features in terms of normalized correlation similarity. Then we propose the most consistent (t_1 , t_2)-vector pair through an iterative process of randomly selecting 3 points to form a (t_1 , t_2) pivot for RANSAC and searching for the pivot with the maximum number of inliers.

At phase two, tracking of each lattice point takes place under a 2D Markov Random Field formulation with compatibility functions built from the proposed (t_1 , t_2)-vector pair and texel. The lattice grows outwards from the initial texel locations using the (t_1 , t_2)-vector pair to detect additional lattice points. The tracking is initiated by predicting lattice points using the proposed (t_1 , t_2)-vector pair under the MRF formulation. The inferred locations are further examined; if the image likelihood at a location is high, then that location becomes part of the lattice. However, for robustness, the method avoids setting a hard threshold and uses the region of dominance idea introduced in [6]. This is particularly important since there is no prior information about how many points to expect

in any given image. If the threshold of detecting lattice points is too high, then recall rate suffers.

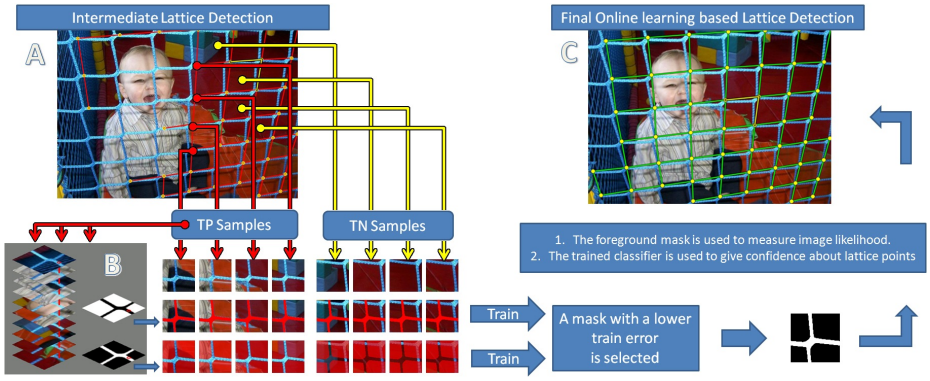


Fig. 2: Procedure of lattice detection using online clustering, learning and classification

Since the performance of lattice detection plays an essential role in this application, we introduce a better decision system that uses online classification and combines the lattice detection procedure with foreground / background segmentation. In addition, we segment out the foreground layer during the detection procedure and build a mask to remove noisy regions of each texel to represent background irregularities from distracting and misguiding the inference procedure. Since evaluation of a noisy image likelihood could misdirect the inference of new texel locations, resulting in inaccurate lattice detection, we evaluate the image likelihood of the each texel by normalized cross correlation using only the foreground mask.

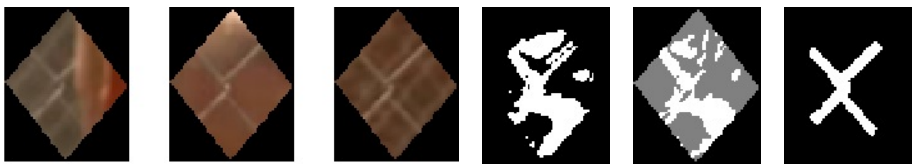
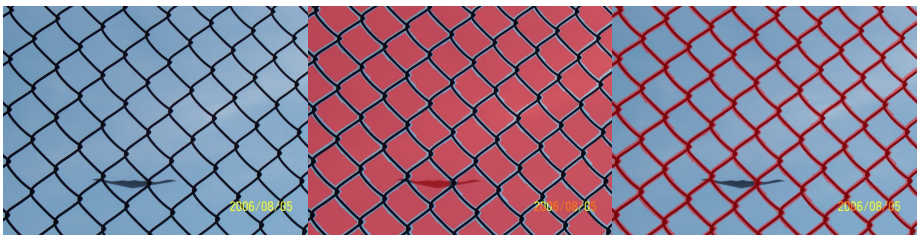


Fig. 3: Sample FG/BG classification for a layer mask. (a) sample texels from the lattice are shown. (b)-(c) results of two methods proposed in [1]. (d) results of our proposed method.

3.1 Clustering for the Foreground Segmentation

Liu et al. [1] simultaneously align multiple texels by calculating a homography for each texel that brings its corners into alignment with the average texel shape (Figure 2 B). After aligning all the texels, they compute the standard deviation of each pixel in each texel with respect to the values at the same location in all other texels. They propose two methods of pixel classification. The first was the classification of background versus foreground by thresholding of the variance among corresponding pixels. The second was to consider the color of each texel along with the variance and performing K-means clustering on 6D vectors composed of the value and standard deviation of red, green, and blue channels for each pixel. They identified the pixels belonging to the lower variance cluster as the lattice region. Sample results from these two methods are shown in Figures 3b and 3c.

Differing from Liu et al. [1], we use the mean of all pixels at each location within the average texel shape. Now the input to the K-means (K=2) clustering is a set of 6D vectors composed of the mean value (for all pixels at that location) and the standard deviation (for each pixel) of red, green, and blue channels. We achieve better results with the use of the mean value, as can be seen in Figure 3d. This is because the means cancel out the irregularities in the backgrounds and make the boundary between the foreground and the background clear.



(a) (b) (c)

Fig. 4: (a) An uniform background can make the relative mean RGB variance of the foreground larger. (b) Results of taking the cluster with lower variance as foreground: red is foreground. (This picture is best viewed in color.) (c) Results of our proposed foreground segmentation.

However, taking the cluster with a smaller RGB variance does not always work since severe lighting conditions on the foreground or a uniform background can result in equivalent RGB variance for each cluster, as can be seen in Figure 4.

3.2 Online Learning-based Lattice Detection

In our lattice detection algorithm, online learning using a support vector machine is performed to improve the classification of lattice points and for foreground seg-

mentation. The base lattice detection algorithm provides both samples, $x_i \in R^n$ and the label of the samples $y_i = \{-1, 1\}$, which enables us to do supervised learning. Positive samples, x_i, $y_i = 1$ are collected from patches centered at lattice points (Figure 2, red arrows). Negative samples x_i, $y_i = -1$ are collected from patch locations between positive samples (Figure 2, yellow arrows). Next, we segment the lattice region to determine the lattice mask using K-means (Section 3.1 and Figure 2 B). At this stage we have two candidates for the foreground mask. Then, at each sample location, RGB color histograms are computed from the two masks and used as features.

We use a support vector machine (SVM) with linear kernel and 10-fold cross validation. We train the SVM to minimize the objective function given by equation (1) with respect to $\mathbf{w}$, b (support vector) and ξ (slack variable for non-separable data). For this purpose, we used the OpenCV Machine Learning toolbox.

$$\begin{array}{c} \min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \\ y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{array} \tag{1}$$

The parameter C (the penalty parameter of the error term in equation (1) and the only optional parameter set by the user for the linear kernel SVM) is iterated on a logarithmic grid and selected based on a 10-fold cross validation estimate of error rate given by the ratio of the number of misclassified samples over the number of test samples. Since we have two possible foreground masks from clustering, we train an optimal classifier for each mask. To decide the best foreground mask for representing the positive samples, x_i, $y_i = 1$ we further examine the collected positive and negative samples using the trained classifiers. The idea behind our approach is that if a mask A is representing x_i, $y_i = 1$ faithfully, then the training error of the classifier with features collected from A should be smaller than that of the classifier with features collected from the other mask, B. We measure the training error of each classifier and select the foreground mask that results in lower training error. The optimal classifier with the selected mask is used to aid further lattice detection, an advancement from [1]. Finally, we consider the foreground mask when determining image likelihood during the lattice point inference procedure, increasing accuracy in localization of lattice points. The procedure repeats until no more texels are found. Our proposed method has a 30% improved detection rate[1] over the state-of-the-art algorithm [5] on the 32 images from the PSU NRT database.

4 Multi-view and Symmetry Augmented Inpainting

One of the most challenging problems in inpainting is the scarcity of source samples [1]. We seek to overcome this in two ways. The first approach is to try

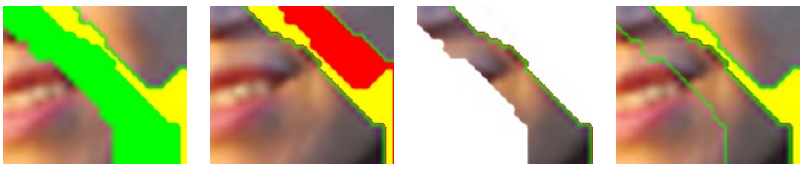
[1] The detection rate is measured by the ratio of the number of correctly detected texels over the total number of ground truth texels.

to see the occluded object in another view. It is reported by Liu et al. [1] that overall occupation of the foreground fence layer in their data set is from 18% to 53%. However, even a small offset of the camera can reveal pixel values behind the foreground layer since objects behind the layer will experience less parallax than the foreground. Also, moving objects will reveal parts of themselves, even to a stationary camera, through multiple frames. Since in video these offsets are small, object alignment can be approximated as a 2D translation. We utilize the information from multiple views to aid the inpainting process by minimizing the number of pixel values that need to be inferred.

A second approach deals with the situation after multi-view inpainting or where no additional views are available. For gaps that still remain, we adopt an exemplar based inpainting algorithm [3][4] as our base tool. In addition ,we seek to overcome scarcity of candidate patches by simulating bilateral symmetry patterns from the source image. As reflection symmetry often exists in man-made environments and nature, simulating these patterns from the source image often recovers occluded regions reliably and efficiently.

4.1 Multi-view Inpainting

To begin fence removal, we first remove the foreground layer (section 3) and then start extracting patches for inpainting. Since the order of synthesis is critical, the method for determining order that appears in [14] is used. That is, any objects that are closer or have moved more between views should be dealt with first because of their depth or motion boundary. Although optical flow estimation is often not robust due to hole regions, errors are generally not noticeable in the resulting image.



(a) Patch 1 (b) Patch 2 (c) Inpainting region (d) Inpainting result

Fig. 5: Process of multiview inpainting and result: The green region shows the region that is made visible by patch 2 and the yellow region shows the region to inpaint using augmented symmetries

For a given image, I, we compute magnitude of optical flow, F, using the Lucas Kanade algorithm [17] for every pixel. The priority of the matching follows a descending order with respect to F. From the location p with the maximum F, we extract patch Φ_p to do block matching with the other view, I'. Formally, we seek to solve,

$$\Phi_{\widehat{q}} = arg \min_{\Phi_q \in I'} SSD(\Phi_p, \Phi_q) \tag{2}$$

where SSD is sum of squared difference.

We use a larger patch width (15∼30) than the original inpainting algorithm (≤ 9) to disambiguate similar patches. This would have posed problems in earlier works [4, 13] because only complete source regions (containing no pixels to be inpainted) were considered as candidates. We allow for an area that matches better to be selected even if some of the pixels of the patch will need to be synthesized later.

Another possible problem of using a larger patch occurs at boundaries between objects at different depths. We attempt to minimize the effect of these depth boundaries by filling in the pixel values in descending order of optical flow magnitude as in [14]. Having found $\Phi_{\widehat{q}}$, the value of each pixel $p \in \Phi_p \cap H$ is copied from its corresponding pixel $q \in \Phi_{\widehat{q}} \setminus H$. If $q(\in \Phi_{\widehat{q}} \setminus H)$ is the null set, the value of p is not observable from any other views, hence we use the single view inpainting algorithm in Section 4.2. As can be seen in Figure 5 we do not replace the entire original patch (Figure 5b), but only replace the region that is occluded in the original patch (Figure 5d).

(a) Normal inpainting (b) Our symmetry-augmented inpainting

Fig. 6: Result of normal inpainting compared with symmetry-augmented inpainting. Both inpainting algorithms are applied after multi-view inpainting. (a) Results of normal inpainting [3, 4] (b) Inpainting with simulated bilateral symmetry patches

4.2 Symmetry-Augmented Inpainting

After multi-view inpainting or when only one view is available, we adopt an exemplar-based single view inpainting algorithm [3, 4] for hole regions that still

remain. As symmetry is common in nature and man-made environments, simulating these patterns from the source image increases the pool of candidate matches, which could improve the inpainting quality. First, size of the template window Φ is given as 9 by 9 for a given image, I, and the patch priority is computed according to [3, 4]. We select the patch with the highest priority, Φ_p and we rotate Φ_p by 90, 180 and 270 degrees as well as flip Φ_p around the x, y, $y = x$ and $y = -x$ axes. We next search in the source region, $S = I \setminus H$ for the patch most similar to Φ_p or its simulated symmetry patches, $\Phi_p^{(i)}$, where $i = 1 \sim 7$. Formally we seek to solve,

$$\Phi_{\widehat{q}} = arg \min_{\Phi_q \in S, i=1\sim 7} SSD(\Phi_p^{(i)}, \Phi_q) \tag{3}$$

Having found the source exemplar $\Phi_{\widehat{q}}$, we apply the appropriate inverse rotation or reflection on $\Phi_{\widehat{q}}$ depending on the index, i, then the value of each pixel $p \in \Phi_p \cap H$ is copied from its corresponding location in $\Phi_{\widehat{q}}$.

As can be seen in Figure 6, although there are still artifacts, our proposed method offers improvements in keeping the image structure (inner corner of sunglasses in Figure 6).

5 Experimental Results

We first compare our method of lattice detection to [5]. We then compare our overall system with [1] on the same images that appeared in [1]. Last, we demonstrate results of multiview and symmetry-augmented inpainting on multiview images.

5.1 Lattice Detection

We have tested on 32 images from the PSU NRT database[2] [18, 19] and have found a 30% improvement in detection rate over [5]. Quantitative evaluation of true positive rate and false positive rate are shown in Table 1. The true positive rate is computed by the ratio of the number of correctly identified texels over the number of ground truth texels, and the false positive rate is computed by the ratio of the number of incorrectly identified texels over the number of ground truth texels. The ground truth data and automatic evaluation code is obtained from the PSU Near Regular Texture Database[2].

5.2 Comparison with Liu et al. [1]

Our proposed method is successful at finding lattices and corresponding masks for both of the images that appeared in [1]. Sample results of [1] and our results[3] are shown in Figure 7.

[2] "http://vision.cse.psu.edu/data/data.shtml".

[3] More results can be found in "http://vision.cse.psu.edu/research/Defencing-Revisited/index.shtml".

Lattice Detection Rate	True Positive	False Positive
Park et al.	$59.34\% \pm 25.58$	$0.62\% \pm 2.4$
Ours	$77.11\% \pm 16.24$	$0.74\% \pm 2.5$

Table 1: Quantitative evaluation of true positive rate and false positive rate, the true positive rate is computed by the ratio of the number of correctly identified texels over the number of ground truth texels and the false positive rate is computed by the ratio of the number of incorrectly identified texels over the number of the ground truth texels

5.3 Multi-view Inpainting Result

We apply our multi-view inpainting and symmetry augmented inpainting to images that have multiple views and a few frames extracted from the show "Prison Break". The results are illustrated in Figure 8. In Figure 8, the first row uses 4 views, the second row uses 3 views, and the last row uses 2 views.

6 Conclusion

We introduce a novel technique for "image de-fencing", the automatic removal of foreground fence layer in real photos, by detecting, segmenting and inpainting repeating foreground structures. We treat detection and segmentation of the lattice as a coupled learning process since the results of each one can be fed to the other to improve the overall performance. Our lattice detection method produces improved results over the state-of-the-algorithm [5] by 30%. We also propose multi-view inpainting and symmetry-augmented inpainting methods to overcome the problem of candidate sample patch impoverishment for inpainting. Even for human faces, these new alternatives lead to acceptable results (Figure 8). Our future goal is to deal with large view angle changes between multiple views.

Acknowledgement This work is supported in part by an NSF grant IIS-0729363 and a gift grant to Dr. Liu from the Northrop Grumman Corporation.

References

1. Liu, Y., Belkina, T., Hays, J., Lublinerman, R.: Image de-fencing. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, Anchorage, Alaska (2008) 1–8
2. 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
3. Criminisi, A., Perez, P., Toyama, K.: Object removal by exemplar-based inpainting. In: CVPR. (2003) 721–728

(a) Liu et al. [1] (b) Ours

Fig. 7: Sample results of Liu et al. [1] and our approach. The middle column shows the results of our proposed segmentation method and the last column shows the results of inpainting. The results show that inpainting using a single view is still very challenging even with a good segmentation. More results can be found in "http://vision.cse.psu.edu/research/Defencing-Revisited/index.shtml".

4. Criminisi, A., Perez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Transactions on Image Processing **13** (2004) 1200–1212
5. Park, M., Collins, R.T., Liu, Y.: Deformed Lattice Discovery via Efficient Mean-Shift Belief Propagation. In: 10th European Conference on Computer Vision, Marsellie, France (2008)
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. Leung, T., Malik, J.: Detecting, localizing and grouping repeated scene elements from an image. In: 4th European Conference on Computer Vision. (1996) 546–555
8. Schaffalitzky, F., Zisserman, A.: Geometric grouping of repeated elements within images. In: Shape, Contour and Grouping in Computer Vision. (1999) 165–181
9. Han, J., McKenna, S., Wang, R.: Regular texture analysis as statistical model selection. In: 10th European Conference on Computer Vision, Marsellie, France (2008)
10. 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
11. Leonard G. O., H., Takeo, K.: Computer analysis of regular repetitive textures. In: Proceedings of a workshop on Image understanding workshop, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1989)
12. Favaro, P., Grisan, E.: Defocus inpainting. In: Computer Vision ECCV 2006. Volume 3952 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2006) 349–359
13. Wexler, Y., Shechtman, E., Irani, M.: Space-time completion of video. Pattern Analysis and Machine Intelligence, IEEE Transactions on **29** (2007) 463–476

(a) Input (b) Liu et al. [1] (c) Multi-view Inpainting

Fig. 8: (a) Input images (b) Results of [1] using a single view. (c) Results of our proposed multiview and symmetry-augmented inpainting method. The 1^{st} row, 2^{nd} row, and 3^{rd} row in (c) uses 4, 3, and 2 views respectively. More results can be found in "http://vision.cse.psu.edu/research/Defencing-Revisited/index.shtml".

14. Patwardhan, K.A., Sapiro, G., Bertalmio, M.: Video inpainting of occluding and occluded objects. In: Image Processing, 2005. ICIP 2005. IEEE International Conference on. Volume 2. (2005) II–69–72
15. James, H., Alexei, A.E.: Scene completion using millions of photographs (2007) 1276382 4.
16. Vaish, V., Levoy, M., Szeliski, R., Zitnick, C.L., Sing Bing, K.: Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures. In: Computer Vision and Pattern Recognition, 2006. Volume 2. (2006) 2331–2338
17. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI) (1981) 674–679
18. Park, M., Lee, S., Chen, P.C., Kashyap, S., Butt, A.A., Liu, Y.: Performance evaluation of state-of-the-art discrete symmetry detection algorithms. In: Proceedings of CVPR 2008. (2008)
19. Chen, P.C., Hays, J.H., Lee, S., Park, M., Liu, Y.: A quantitative evaluation of symmetry detection algorithms. Technical Report CMU-RI-TR-07-36, Robotics Institute, Pittsburgh, PA (2007)