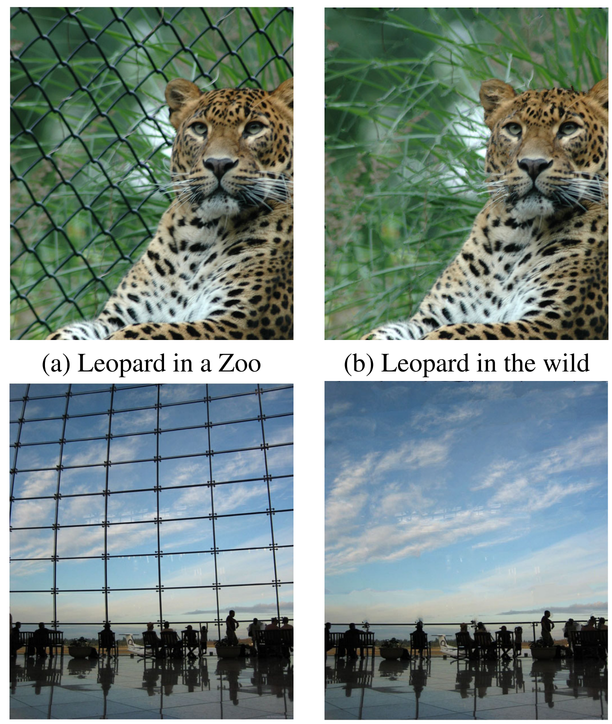
Image De-fencing

Yanxi Liu[†,*], Tamara Belkina[†], James H. Hays[+], and Roberto Lublinerman[†]
[†]Department of Computer Science and Engineering, [*]Department of Electrical Engineering
The Pennsylvania State University
[+]Computer Science Department, Carnegie Mellon University
{yanxi,belkina,lubliner}@cse.psu.edu, hays@cs.cmu.edu, http://vision.cse.psu.edu/defencing.htm

Abstract

We introduce a novel image segmentation algorithm that uses translational symmetry as the primary foreground/background separation cue. We investigate the process of identifying and analyzing image regions that present approximate translational symmetry for the purpose of image fourground/background separation. In conjunction with texture-based inpainting, understanding the different see-through layers allows us to perform powerful image manipulations such as recovering a mesh-occluded background (as much as 53% occluded area) to achieve the effect of image and photo de-fencing. Our algorithm consists of three distinct phases– (1) automatically finding the skeleton structure of a potential frontal layer (fence) in the form of a deformed lattice, (2) separating foreground/background layers using appearance regularity, and (3) occluded foreground inpainting to reveal a complete, non-occluded image. Each of these three tasks presents its own special computational challenges that are not encountered in previous, general image de-layering or texture inpainting applications.



(a) Leopard in a Zoo (b) Leopard in the wild

(c) People in an airport (d) People on a deck

Figure 1. (a) and (c) are real world photos with a foreground near-regular layer. (b) and (d) are recovered background of the input photos using our detection-classification-inpainting algorithm.

1. Introduction

We address a novel problem of detecting, segmenting, and inpainting repeating structures in real photos (Figure 1). The understanding of the different image layers coupled with texture-based inpainting allows us to perform useful image manipulations such as recovering a heavily occluded background from a foreground occluder that occupies the majority of the image area. The novelty of our application is to computationally manipulate this type of space-covering, fence-like, near-regular foreground patterns that are oftentimes unwanted and unavoidable in our digital imagery. Common examples include zoo pictures with animals in their respective wired cages (Figure 1 (a)), children's tennis/baseball games that can only be observed behind fences, reflections of near-by buildings, shadows of frames (Figure 4), or fantastic views that can only be watched through a set of glass windows (Figure 1 (c)).

Traditional texture filling tools such as Criminisi et al. [4] require users to manually mask out unwanted image regions. Based on our own experience, for images such as those in Figure 1 this process would be tedious (taking hours) and error-prone. Simple color-based segmentations are not sufficient. Interactive foreground selection tools such as Lazy Snapping [12] are also not suited to identifying these thin, web-like structures. Painting a mask man-

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

ually, as in previous inpainting work, requires copious time and attention because of the complex topology of the foreground regions. Effective and efficient photo editing tools that can help to remove these distracting, but unavoidable, near-regular layers are desirable. By providing such image editing tools, a user will have the capability to reveal the most essential content in a photo without unwanted intrusions (Figure 1).

These repeated image structures can be described as near-regular textures which are deformations from regular, wallpaper-like patterns [17]. Near-regular textures possess an underlying lattice– a space-covering quadrilateral mesh that unambiguously specifies the topology of the texture and allows all texels (unit tiles) to be put into accurate correspondence with each other. Since our goal in this paper is de-layering, the near-regular textures we encounter are usually embedded in irregular backgrounds (Figure 1 (b) and (d)).

We propose an image de-fencing algorithm consisting three distinct phases: (1) automatically finding the skeleton structure of a potential frontal layer in the form of a deformed lattice; (2) classifying pixels as foreground or background using appearance regularity as the dominant cue, and (3) inpainting the foreground regions using the background texture which is typically composed of fragmented source regions to reveal a complete, non-occluded image. The three phases are intertwined with non-trivial computational constraints. Each of these three tasks presents its own special computational challenges that are not encountered in previous general image de-layering or texture inpainting. Our results demonstrate the promise as well as great challenges of image de-fencing.

One basic assumption in this work is that if each layer in a 3D scene is associated with certain level of regularity, then the most regular layer that can be observed in its completeness is the frontal layer. The argument for the validity of this assumption is simply: otherwise (if the regular layer is occluded by something else), its regularity will no longer be observed. This observation is also consistent with the Gestalt principles of perception that stress the importance of perceptual grouping using symmetry. Different from contour and continuity cues which are also emphasized in Gestalt theory, the regularity or translational symmetry cues we use are discrete. It is precisely owing to this non-continuous nature that this discrete symmetry or regularity cue can lead our algorithm through cluttered scenes to segment out and distill the 'hollow', repeated patterns that have similar pixels only in certain discrete, corresponding regions. In a way, the structure (the lattice) of the foreground is being lifted from the image to form the bases for our foreground mask regions.

Our work differs in several aspects from existing work. First of all, unlike classic inpainting where the user provides masks, phase I of our algorithm automatically discovers the deformed lattice that characterizes the translationally repeated pattern and in phase II the mask is automatically generated from the lattice according to local color and global color variations on *corresponding* pixels. Secondly, different from Hays et al. [9] where the main goal is to find those near-regular textures that share locally similar texels, the focus in our current work is to discover the non-obvious, fence-like patterns placed over drastically varying background textures (e.g. Figure 1). Last but not least, the source texture region is seriously fragmented compared with existing inpainting work that is usually done using large, continuous areas of source textures. Furthermore, the ratio of foreground (to be filled) area to background area in our work is much higher than the usual 10-15% previously reported[4]. The ratios in our work range from 18% to 53%. All these differences pose extra computational challenges.

The contribution of this work includes the identification of a new and real application sitting on the boundary of computer vision and computer graphics: image de-fencing; a unique translation-symmetry based foreground/background classifier; a demonstration of the promising results from our proposed automatic detection-classification-inpainting procedure; and the discovery of the limitations of both the lattice detection [9] and the popular inpainting algorithms [4] for photos with near regular texture patterns. We demonstrate both success and failure examples at different stages of image de-fencing, emphasizing the fact that automatic image de-fencing, for the most part, remains an unsolved novel problem.

2. Related work

2.1. Finding Image Regularity

There is a long history of research concerning the identification of regular and near-regular patterns in images. Notable, recent works include Tuytelaars et al. [25] and Schaffalitzky et al. [22] identify regular patterns under perspective distortion. Leung and Malik [11] finds connected, visually similar structures in photographs under arbitrary distortion. Forsyth [7] finds and clusters visually similar, foreshortened textons with less regard for their geometric arrangement. Liu et al. [16] present a robust method for finding the lattice of regular textures. We employ Hays et al. [9] to automatically find near-regular lattices in real world images.

Liu et al. [17] demonstrated several image regularity manipulations based on knowing a lattice of a near-regular texture but required the lattice to be specified, in part, interactively. Tsin et al. [24] and Liu et al. [17] both use regularity as a cue to perform texture synthesis, manipulation and replacement using real photos. Lin and Liu [13, 14] and White and Forsyth [26] extend this to video.

Our work also attempts to replace a near-regular texture but our goals are different. Previous work replaces a near-regular texture with an arbitrary, user-provided texture while preserving lighting and curvature regularity thus giving the appearance of a new texture on the exact same surface. We are selectively replacing a 'partial' near-regular texture with the surrounding irregular texture to give the impression that the regular surface never existed (Figure 1).

2.2. Photo Manipulation

With the growth of digital imaging there has been considerable research interest in intelligent photo processing. For instance, [2, 6, 19] all deal with enhancing or correcting photographs involving flash, for example, [8] deals with correcting red-eye automatically. These types of artifacts are common but relatively easy to mitigate at capture time with indirect flash lighting. But in some situations it might be impossible to avoid the types of occlusions we segment and remove. While those papers aim to correct artifacts that appear because of the imaging and lighting process, Agarwala et al. [1] describes an interactive framework in which higher-level photographic manipulations can be performed based on the fusion of multiple images. In addition to repairing imaging artifacts, users can interactively improve composition or lighting with the help of multiple exposures. In a similar vein, our method can correct the unfortunately occlusions in photo composition such as a subject behind a fence. In many situations, such as a zoo, these types of compositions may be unavoidable.

2.3. De-layering

This work can be viewed as solving the figure/ground image labelling problem based on a single, strong cue– regularity, rather than an ensemble of cues as in [20]. In [3] and [15] video is automatically decomposed into distinct layers based on the occlusion of moving objects and the grouping of coherent motions. Defocus Video Matting [18] distinguishes foreground and background automatically based on multiple, differently focused but optically aligned images. GrabCut [21] and Lazy Snapping [12] are user assisted, graph-cut based methods for segmenting foreground from background. Our approach is one of the few methods aimed at foreground/background segmentation based on a single image.

3. Approach

Our method has three distinct yet inter-related phases– 1) Finding a lattice 2) Classifying pixels as foreground or background 3) Filling the background holes with texture inpainting.

3.1. Finding a Lattice

In order to understand the regularity of a given image we seek a lattice which explains the relationship between repeated elements in a scene. We use the method implemented in [9], which is an iterative algorithm trying to find the most regular lattice for a given image by assigning neighbor relationships among a set of interest points, and then using the strongest cluster of repeated elements to propose new, visually similar interest points. The neighbor relationships are assigned such that neighbors have maximum visual similarity. More importantly, higher-order constraints promote geometric consistency between pairs of assignments. Finding the optimal assignment under these second-order constraints is NP-hard so a spectral approximation method [10] is used.

No specific restrictions on the type of visual or geometric deformations present in a near-regular texture are imposed in [9], but with increasing irregularity it becomes more difficult to find a reasonable lattice. When a see-through regular structure is overlaid onto an irregular background, such as in our examples (Figures 1,4,5, and 6), finding a lattice is especially challenging. If the regularity is too subtle or the irregularity too dominant the algorithm will not find potential texels. To alleviate this we lower the threshold for visual similarity used in [9] for the proposal of texels. Since the near-regular structures in our test cases tend to have relatively small geometric deformations, the algorithm can still find the correct lattice even with a large number of falsely proposed texels that might appear with a less conservative texel proposal method.

The final lattice is a connected, space-covering mesh of quadrilaterals in which the repeated elements contained in each quadrilateral (hereafter 'texels') are maximally similar. The lattice does not explicitly tell us which parts of each texel are part of a regular structure or an irregular background. However, the lattice does imply a dense correspondence between all texels which allows us to discover any spatially distinct regular and irregular subregions of the texels which correspond to foreground and background respectively.

3.2. Foreground/background separation

We put all of our texels into correspondence by calculating a homography for each texel which brings the corners into alignment with the average-shaped texel. After aligning all the texels we compute the standard deviation of each pixel through this stack of texels (Figure 2). We could classify background versus foreground based on a threshold of this variance among corresponded pixels but a more accurate classification is achieved, when we consider color information in each texel in addition to their aggregate statistics. We couple each pixel's color with the standard de-

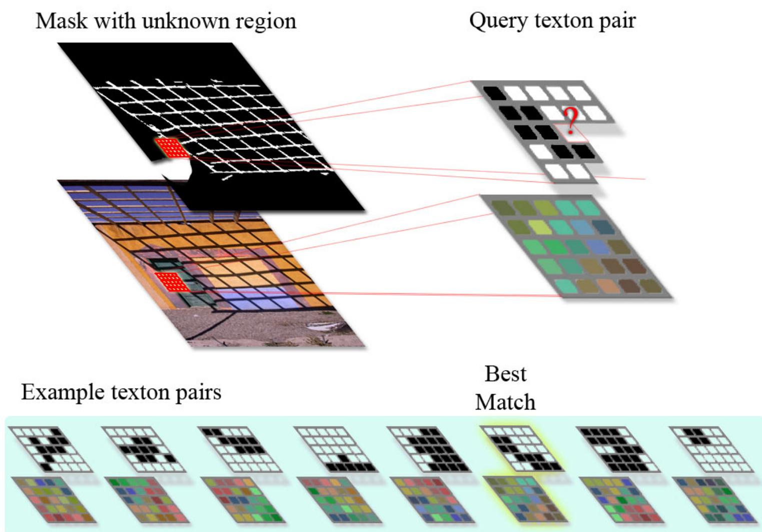


Figure 3. Unknown regions of the mask are filled in one pixel at a time by finding the most similar mask and image pair in the already determined regions. For a particular query pair (upper right) distance is computed to all labeled pairs of textons. The best match for this particular texton is highlighted, and the center pixel of the query texton's mask will take the value from this best match.

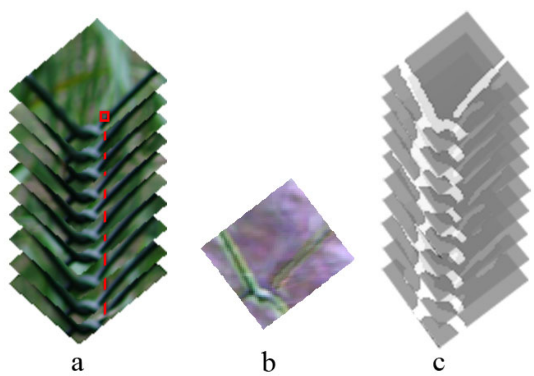


Figure 2. (a) A stack of aligned texels. (b) Standard deviation is calculated along each vertical column of pixels. (c)The standard deviation and color of all pixels is clustered to discover foreground and background.

viation of each color channel at its offset in the aligned texel. This gives us as many 6-dimensional examples as we have pixels within the lattice. There is a constant, relative weighting between the standard deviation and the color features for k-means. The standard deviation is weighted more heavily. Different values of k are used for k-means for different examples, from 2 to 4. We cluster these examples with k-means and assign whichever cluster ends up having the lowest variance centroid to be the foreground and the rest background. From this classification we construct a 'mask' in image space which corresponds to foreground, background, and unknown.

3.3. Image De-fencing – Background Texture Fill

We can estimate a plausible background by applying texture based inpainting to all pixels which have been labeled as foreground. We use the method of Criminisi et al. [4], which modifies Efros and Leung [5] by changing the order in which pixels are synthesized to encourage continuous, linear structures. The modified synthesis order profoundly improves inpainting results even for regions that are relatively thin such as ours. Patch-based image completion methods[23] are less appropriate for our inpainting task because our target regions are perhaps a single patch wide which obviates the need for sophisticated patch placement strategies as explored in [23]. Also our source regions offer few complete patches to draw from. On the other end of the inpainting spectrum, diffusion-based inpainting methods also work poorly. Our target regions are wide enough such that the image diffusion leaves obvious blurring.

A mask which appears to cover a foreground object perfectly can produce surprisingly bad inpainting results due to a number of factors: the foreground objects are often not well focused because our scenes often have considerable depth to them, sharpening halos introduced in post-processing or in the camera itself extend beyond the foreground object, and compression artifacts also reveal the presence of an object beyond its boundary. All of these factors can leave obvious bands where a foreground object is removed. In order to remove all traces of a foreground object we dilate our mask considerably before applying inpainting.

Our inpainting task is especially challenging compared to previous inpainting work by [4]. We typically have a

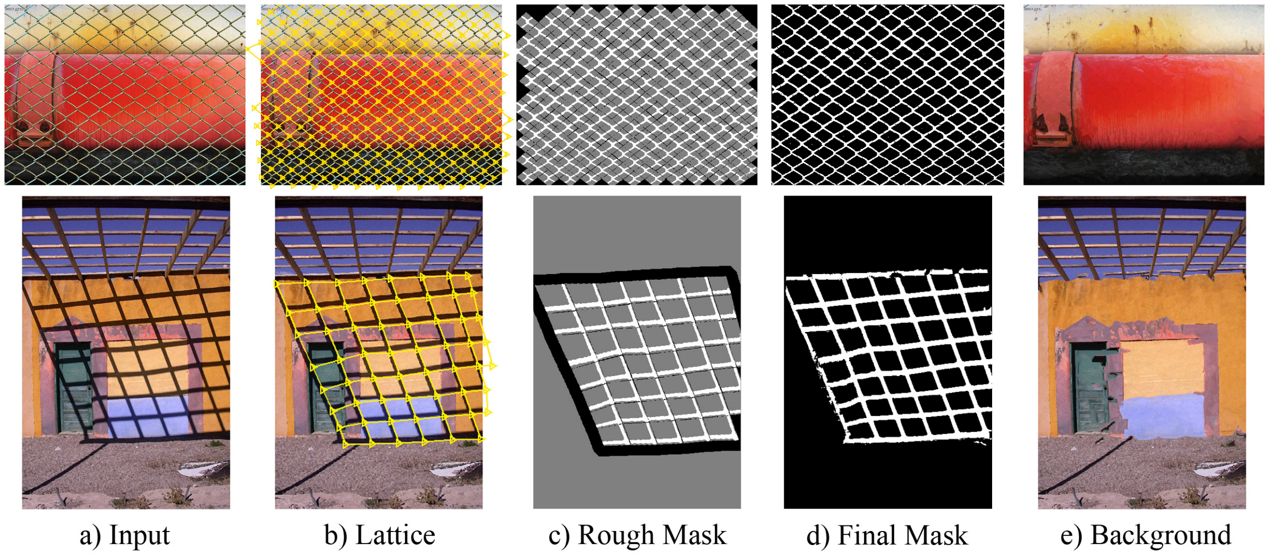


Figure 4. The procedure of de-fencing is shown step-by-step through two examples, where one instance of the 'fence' is composed of shadow.

large percentage of the image to fill in, from 18 to 53 percent after dilation (about 3 pixels per image, e.g. the pipe image was 53% masked out Figure 4), and the boundary between our source regions and our masked-out regions has a very large perimeter. These factors conspire to give us few complete samples of source texture with which to perform the inpainting - a new problem rarely happened in previous inpainting applications where large regions of source textures with simple topology are available.

4. Experimental Results

We have tested our algorithm on a variety of photos obtained from the Internet. Figure 4 shows various stages of the detection-classification-inpainting process. First a lattice is found using [9] (4b). Our novel foreground/background classifier then uses the amount of regularity in the aligned texels to compute a rough mask for the pixels covered by the lattice (4c). We extend this classification into nearby regions (4d) and fill the foreground regions with texture inpainting (4e).

Figure 5 shows some promising results, including the intermediate results of the photos shown in Figure 1 (In 1a the repeated structure is itself occluded by another object); while Figures 6 and 7 show failure results at lattice detection and image inpainting stages respectively. A total of 44 images with various fence-like structures are tested, 31 of them failed to obtain a complete lattice at the lattice detection step (Figure 6). For those with correctly detected lattice, 6 images are left with much to be desired in their inpainting results (Figure 7).

5. Discussion and Conclusion

We have introduced and explored the novel use of translational symmetry for image de-fencing and photo-editing with inpainting. The results (Figures 4 and 5) demonstrate the plausibility of using regularity as a foreground/background segmentation cue. A near-regular structure can be automatically detected and segmented out of cluttered background.

Automatic lattice detection from real images [9] has met some serious challenges in this application: detection of see-through, near-regular structures from adverse background clutters. We have observed the failure cases (Figure 6) often are accompanied by sudden changes of colors in the background (e.g. peacock, moose); obscuring objects in front of the fence (e.g. building), and irregular background geometry.

Based on our experimental results, and contrary to our initial expectations, we observe that the mesh-like regions are actually more difficult to texture fill than large, circular regions of similar area. This is because the mesh-like regions are wide enough to show errors with incorrect structure propagation, but they have dramatically larger perimeter than a single large region and thus there are many more structures which need to be correctly propagated and joined. Mistakes in structure propagation can be seen in our results such as the shadowed wall in Figure 4e. The fragmentation of the source regions caused by the complex topology of the regular structures is also problematic: there are no long, consecutive texture strips for the texture filling algorithm to use so the texture filling is forced to have low coherence and thus the quality of inpainting suffers. The high ratio of foreground area to background area as well as the fragmented

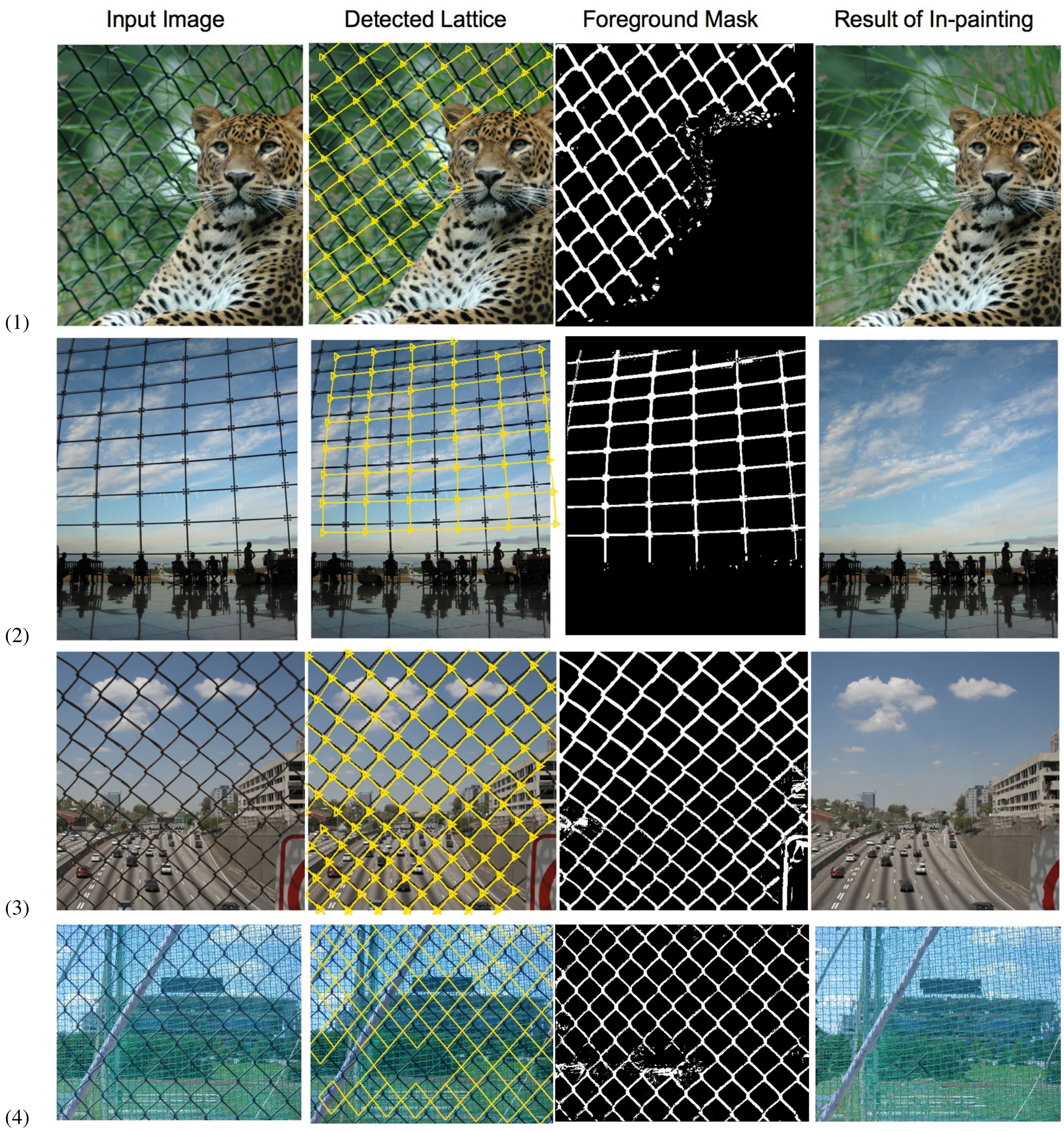


Figure 5. Several relatively promising image de-fencing results demonstrate the effectiveness of the proposed, translation symmetry-based detection-classification-inpainting method. The lattice detection in (4) is imperfect, thus a piece of the fence remains after inpainting.

background source textures present special challenges for existing inpainting methods. Further study on how to improve the state of the art inpainting methods to suit this type of source-texture-deprived situations will lead to more fruitful results.

References

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. *ACM Transactions on Graphics (SIGGRAPH)*, 23(3):294–302, 2004.

Figure 6. Examples where the lattice detection algorithm [9] failed to extract the complete, correct lattice in each image.

[2] A. Agrawal, R. Raskar, S. Nayar, and Y. Li. Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Transactions on Graphics (SIGGRAPH)*, 24(3):828–835, 2005.

[3] G. J. Brostow and I. Essa. Motion based decompositing of video. In *IEEE International Conference on Computer Vision (ICCV)*, pages 8–13, 1999.

[4] A. Criminisi, P. Prez, and K. Toyama. Region filling and object removal by exemplar-based inpainting. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2003.

[5] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1033–1038, 1999.

[6] E. Eisemann and F. Durand. Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics (SIGGRAPH)*, 23(3):673–678, 2004.

[7] D. A. Forsyth. Shape from texture without boundaries. In *Proc. European Conf. Computer Vision (ECCV)*, pages 225–239, 2002.

[8] M. Gaubatz and R. Ulichney. Automatic red-eye detection and correction. In *ICIP 2002: IEEE International Conference on Image Processing*, pages 804–807, 2002.

[9] J. Hays, M. Leordeanu, A. Efros, and Y. Liu. Discovering texture regularity as a higher-order correspondence problem. In *European Conference on Computer Vision (ECCV'06)*, 2006.

[10] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *IEEE International Conference on Computer Vision (ICCV)*, 2005.

[11] T. K. Leung and J. Malik. Detecting, localizing and grouping repeated scene elements from an image. In *Proc. European Conf. Computer Vision (ECCV)*, pages 546–555, 1996.

[12] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM Transactions on Graphics (SIGGRAPH)*, 23(3):303–308, 2004.

[13] W. Lin and Y. Liu. Tracking dynamic near-regular textures under occlusion and rapid movements. In *9th European Conference on Computer Vision (ECCV'06), Vol(2)*, pages 44–55, 2006.

[14] W. Lin and Y. Liu. A lattice-based mrf model for dynamic near-regular texture tracking. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 29(5):777–792, May 2007.

[15] C. Liu, A. Torralba, W. T. Freeman, F. Durand, and E. H. Adelson. Motion magnification. *ACM Transactions on Graphics (SIGGRAPH)*, 24(3):519–526, 2005.

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

[17] Y. Liu, W. Lin, and J. Hays. Near-regular texture analysis and manipulation. *ACM Transactions on Graphics (SIGGRAPH)*, 23(3):368–376, August 2004.

[18] M. McGuire, W. Matusik, H. Pfister, J. F. Hughes, and F. Durand. Defocus video matting. *ACM Transactions on Graphics (SIGGRAPH)*, 24(3):567–576, 2005.

[19] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash

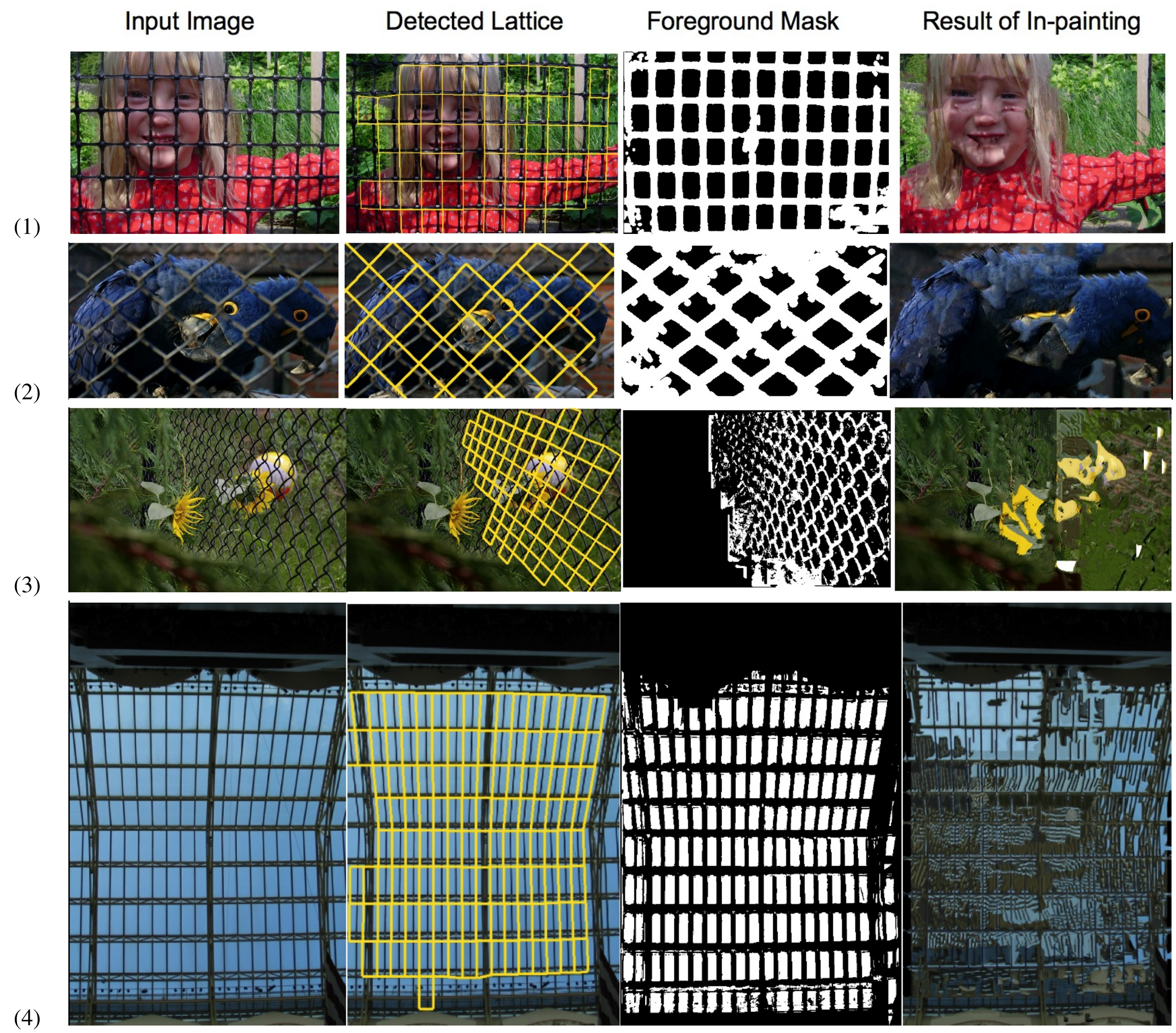


Figure 7. (1)-(3): Image de-fencing failed at the inpainting stage. (4): since the background (sky) is more regular than the foreground, the sky is identified as the 'hole' to be filled during inpainting.

and no-flash image pairs. *ACM Transactions on Graphics (SIGGRAPH)*, 23(3):664–672, 2004.

[20] X. Ren, C. C. Fowlkes, and J. Malik. Cue integration in figure/ground labeling. In *Advances in Neural Information Processing Systems 18*, 2005.

[21] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (SIGGRAPH)*, 23(3):309–314, 2004.

[22] F. Schaffalitzky and A. Zisserman. Geometric grouping of repeated elements within images. In *Shape, Contour and Grouping in Computer Vision*, pages 165–181, 1999.

[23] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum. Image completion with structure propagation. *ACM Transactions on Graphics (SIGGRAPH)*, 24(3):861–868, 2005.

[24] Y. Tsin, Y. Liu, and V. Ramesh. Texture replacement in real images. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01).*, pages 539–544, Kauai, December 2001. IEEE Computer Society Press.

[25] T. Tuytelaars, A. Turina, and L. Van Gool. Noncombinatorial detection of regular repetitions under perspective skew. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, 25(4):418–432, 2003.

[26] R. White and D. A. Forsyth. Retexturing single views using texture and shading. In *Proc. European Conf. Computer Vision (ECCV)*, 2006.