SIViP (2016) 10:1193–1201
DOI 10.1007/s11760-016-0876-7

ORIGINAL PAPER

Image de-fencing framework with hybrid inpainting algorithm

Muhammad Shahid Farid[1,2] · Arif Mahmood[3] · Marco Grangetto[1]

Received: 12 November 2014 / Revised: 8 February 2016 / Accepted: 13 February 2016 / Published online: 2 March 2016

Abstract Detection and removal of fences from digital images become essential when an important part of the scene turns to be occluded by such unwanted structures. Image de-fencing is challenging because manually marking fence boundaries is tedious and time-consuming. In this paper, a novel image de-fencing algorithm that effectively detects and removes fences with minimal user input is presented. The user is only requested to mark few fence pixels; then, color models are estimated and used to train Bayes classifier to segment the fence and the background. Finally, the fence mask is refined exploiting connected component analysis and morphological operators. To restore the occluded region, a hybrid inpainting algorithm is proposed that integrates exemplar-based technique with a pyramid-based interpolation approach. In contrast to previous solutions which work only for regular pattern fences, the proposed technique is able to remove both regular and irregular fences. A large number of experiments are carried out on a wide variety of images containing different types of fences demonstrating the effectiveness of the proposed approach. The proposed approach is also compared with state-of-the-art image de-fencing and inpainting techniques and showed convincing results.

Keywords Image de-fencing · Object removal · Fence removal · Image completion · Hybrid inpainting

✉ Muhammad Shahid Farid
farid@di.unito.it

[1] Dipartimento di Informatica, Università degli Studi di Torino, Torino, Italy

[2] Punjab University College of Information Technology, University of the Punjab, Lahore, Pakistan

[3] Department of Computer Science and Engineering, College of Engineering, Qatar University, Doha, Qatar

1 Introduction

Image de-fencing is an important problem in recreational photography where seamless removal of fences from the photographs is required due to esthetic reasons. The removal should be undetectable to the general viewers. Many times, a photographer captures scenes that are behind fences such as wild animals in cages and natural scenes behind barbed fences. These fences may spread over the entire image, and their manual removal can be very tedious, time-consuming and generally requires special artistic skills. Therefore, the semiautomatic and user-friendly image de-fencing proposed in this paper turns out to be an important tool in computational photography.

Image de-fencing is challenging because automatic fence detection is a difficult task. Fences can be of different shapes, texture and color. Some fences have regular repeating patterns, while others are completely irregular. Moreover, the same fence may also have variations in color, texture and shape. To the best of our knowledge, currently no automatic fence detection technique exits that may work for both regular and irregular fences. Secondly, a good restoration algorithm is required that seamlessly replaces the fence pixels with an estimate of the occluded background pixels. Image inpainting techniques may serve the purpose if the fence region is small and coherent. In the presence of large fences, the existing image inpainting techniques may not get a good estimate of the occluded pixels.

The term 'image de-fencing' is due to Liu et al. [17]. Their fence detection approach is based on the observation that fences often have a regular or near-regular repeating patterns (lattice) such as diamonds, rectangles or squares. The algorithm proposed in [14] is used to detect the lattice. The fence region is extracted and inpainted using texture-based inpainting proposed by Criminisi et al. [9]. The technique proposed in [17] exhibits good results when the fence is quasi-regular, while it exhibits severe detection errors on irregular fences. The authors show that their detection algorithm may fail [17] even on some near-regular pattern fence. Moreover, the fence removal procedure does not produce plausible results, especially when the region covered by the fence is relatively large. The image de-fencing technique proposed in [24] improved the automatic fence detection and also proposed to use multiple views of the scene to inpaint occluded regions. However, the single-view images are inpainted using [9]. Similar algorithms have been proposed to detect the near-regular patterns with user intervention [15,16,18,19,29].

The current image de-fencing algorithms rely on the already existing image inpainting techniques [3,25,27] to restore the fence region. The inpainting technique proposed in [8,9] is considered to be a seminal work that introduced the so-called *exemplar-based or patch-based inpainting*. It used exemplar-based texture synthesis to replicate the texture and structure in the occluded region. The algorithm proposed in [20] improves the exemplar inpainting by exploiting the depth map of the color image. Image inpainting [21] used single Gaussian diffusion kernel to restore the damaged region. It is observed that a damaged region may not have same width across the region. In such cases, using a single diffusion kernel may not produce good results. The inpainting technique proposed in [10] automatically uses weighted kernels of variable size to restore the damaged regions with changing width. However, this technique is also limited to recover small regions. An algorithm for removing stains from images is proposed in [32], which restores the structure and texture in the image. Several other techniques [2,4–6,22,30,31] have also been proposed, and most existing techniques produce good results for small occluded regions, while few improved inpainting algorithms such as [1,7] show better results in case of large coherent regions.

We propose a semiautomatic fence detection algorithm that can detect regular and irregular pattern fences of varying shapes, textures and colors with significantly higher accuracy than the existing algorithms. The proposed fence detection technique computes the statistical characteristics of the fence pixels marked by a user and use them to extract the fence. To improve fence segmentation accuracy, we use context aware morphological techniques. To restore the fence region, we propose a hybrid inpainting technique by blending patch-based inpainting [9] with image pyramid-based inpainting [11] to produce improved visual results. The results of the proposed image de-fencing technique are compared with the existing state-of-the-art techniques on a wide variety of images containing various types of fences. Objective metrics are also provided to measure the quality of the restored images. Our conclusion is that the proposed technique significantly outperforms the existing fence detection and restoration techniques.

The rest of the paper is organized as follows: the proposed fence detection technique is described in Sect. 2. The fence refinement procedure is presented in Sect. 3 and fence restoration in Sect. 4. The experimental results and objective quality assessment are provided in Sect. 5, and the conclusions are drawn in Sect. 6.

2 The proposed fence detection algorithm

The proposed de-fencing algorithm works in four steps, starting with the estimation of the fence color model, which is used in the second step to segment the fence. In the third step, the fence mask is refined by eliminating the false positives and false negatives, and finally the fence region is recovered by a novel hybrid inpainting algorithm.

2.1 Fence pixels classification

It can be noted that most fences share certain statistical characteristics, which can be used to identify the fence pixels in the image. One of such characteristics is the fence color. Indeed, the range of the fence color may be very limited; therefore, the user can select a small number of fence pixels ($10 \approx 15$), which are then used for fence modeling (other methods of selection can be used to specify fence pixels sample like drawing a line on the fence or coloring the fence with paint brush). To minimize the selection error and for increasing the data sample for better modeling accuracy, the k-neighbors of any manually selected pixel are considered as fence samples. The fence samples are then used to estimate the mean and the covariance matrix, and hence, a Gaussian distribution model of the fence is constructed. We chose Gaussian distribution as it is simple to handle analytically and easy to estimate. Moreover, it is observed that the fence color dispersion is usually very limited making the usage of Gaussian distribution for its modeling both simple and very effective. Using such distribution model, fence pixels in the image are identified.

Let I be an image of size $M \times N$ and $I(x_1, y_1), I(x_2, y_2), \ldots, I(x_n, y_n)$ be the n fence pixels selected by the user. The color components of these n pixels and their k neighbors are represented as a $n(k+1) \times 3$ matrix $\mathscr{P}$. Where the columns of $\mathscr{P}$ contain red, green and blue components, respectively, of the $(k+1)n$ fence sample pixels. Given the sample pixels in $\mathscr{P}$, the mean μ and covariance Σ of each color component can be computed as:

$$\begin{bmatrix} \mu_r = \frac{1}{n(k+1)} \sum_{i=1}^{n(k+1)} \mathscr{P}(i,1) \\ \mu_g = \frac{1}{n(k+1)} \sum_{i=1}^{n(k+1)} \mathscr{P}(i,2) \\ \mu_b = \frac{1}{n(k+1)} \sum_{i=1}^{n(k+1)} \mathscr{P}(i,3) \end{bmatrix}; \quad \sum = \begin{bmatrix} \sigma_{(r,r)} & \sigma_{(r,g)} & \sigma_{(r,b)} \\ \sigma_{(g,r)} & \sigma_{(g,g)} & \sigma_{(g,b)} \\ \sigma_{(b,r)} & \sigma_{(b,g)} & \sigma_{(b,b)} \end{bmatrix}$$

where $\sigma_{(\alpha,\beta)}$ is given by:

$$\sigma_{(\alpha,\beta)} = \frac{1}{n(k+1)} \sum_{i=1}^{n(k+1)} (\mathscr{P}(i,\alpha) - \mu_\alpha)(\mathscr{P}(i,\beta) - \mu_\beta))$$

where $\alpha, \beta \in \{1, 2, 3\}$ and $\mu_1 = \mu_r$, $\mu_2 = \mu_g$ and $\mu_3 = \mu_b$. Based on μ and Σ, we build two classification stages. A first rough classification is based on the well-known Mahalanobis distance. This classification may lack in accuracy if the user input is not precise. Then, the classification is refined using Bayes detection of fence pixels. Mahalanobis distance $d_{t,f}$ of each test pixel $\mathbf{x}_t = [r_t, g_t, b_t]^\top$ is defined as:

$$d_{t,f} = (\mathbf{x} - \mu)^\top (\Sigma)^{-1} (\mathbf{x} - \mu) \tag{1}$$

Each pixel is then classified as fence if its distance is less than a predefined threshold τ, non-fence pixel otherwise. Then, the classification is refined applying Bayes' theorem to compute posterior probability for the fence and the non-fence classes, and pixels are assigned to the class with higher probability. Posterior probability is defined as the Bayesian distance from the class mean and covariance matrix.

Let p_f and p_{nf} be the prior probabilities that a pixel is and is not part of the fence. For a given pixel $\mathbf{x}$, the posterior probability of fence class is given as:

$$P(\mu_f, \Sigma_f|\mathbf{x}) = \frac{P(\mathbf{x}|\mu_f, \Sigma_f)P(\mu_f, \Sigma_f)}{P(\mathbf{x})} \tag{2}$$

where μ_f, Σ_f are mean and covariance for fence class and $P(\mathbf{x}|\mu_f, \Sigma_f)$ is the likelihood that $\mathbf{x}$ belongs to fence class represented by a Gaussian distribution model (μ_f, Σ_f) and is calculated as:

$$P(\mathbf{x}|\mu_f, \Sigma_f) = \frac{1}{(2\pi)^{\frac{3}{2}}\sqrt{|\Sigma_f|}} \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\mu)^\top(\Sigma)^{-1}(\mathbf{x}-\mu)} \tag{3}$$

where $P(\mu_f, \Sigma_f)$ is the fence distribution model given by: $P(\mu_f, \Sigma_f) = p_f$. The $P(\mathbf{x})$ is the normalizing factor and may be computed as:

$$P(\mathbf{x}) = P(\mathbf{x}|\mu_f, \Sigma_f)p_f + P(\mathbf{x}|\mu_{nf}, \Sigma_{nf})p_{nf} \tag{4}$$

where (μ_{nf}, Σ_{nf}) is non-fence class distribution model. Equation (2) can now be written as:

$$P(\mu_f, \Sigma_f|\mathbf{x}) = \frac{\mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\mu)^\top(\Sigma_f)^{-1}(\mathbf{x}-\mu)} p_f}{(2\pi)^{\frac{3}{2}}\sqrt{|\Sigma_f|}P(\mathbf{x})} \tag{5}$$

Analogously, the posterior of a pixel for non-fence class may be computed by:

$$P(\mu_{nf}, \Sigma_{nf}|\mathbf{x}) = \frac{\mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\mu)^\top(\Sigma_{nf})^{-1}(\mathbf{x}-\mu)} p_{nf}}{(2\pi)^{\frac{3}{2}}\sqrt{|\Sigma_{nf}|}P(\mathbf{x})} \tag{6}$$

For a particular pixel $\mathbf{x}$, $P(\mathbf{x})$ is same for $P(\mu_f, \Sigma_f|\mathbf{x}))$ and $P(\mu_{nf}, \Sigma_{nf}|\mathbf{x})$, and $(2\pi)^{\frac{3}{2}}$ is a constant, so these two terms can be ignored to reduce the computational cost. Hence, Eqs. (5) and (6) can be rewritten as:

$$P'(\mu_f, \Sigma_f|\mathbf{x}) = \frac{1}{\sqrt{|\Sigma_f|}} \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\mu)^\top\Sigma_f^{-1}(\mathbf{x}-\mu)} p_f \tag{7}$$

$$P'(\mu_{nf}, \Sigma_{nf}|\mathbf{x}) = \frac{1}{\sqrt{|\Sigma_{nf}|}} \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\mu)^\top\Sigma_{nf}^{-1}(\mathbf{x}-\mu)} p_{nf} \tag{8}$$

For each pixel $\mathbf{x}$ in the image, posterior probabilities of fence and non-fence classes are computed for each $\mathbf{x}$ using Eqs. (7) and (8), respectively; finally, $\mathbf{x}$ is assigned to the class with the highest posterior probability:

$$\begin{array}{ll} \mathbf{x} \in \mathrm{FC} & \text{if } P'(\mu_f, \Sigma_f|\mathbf{x}) \geq P'(\mu_{nf}, \Sigma_{nf}|\mathbf{x}), \\ \mathbf{x} \in \mathrm{NFC} & \text{otherwise} \end{array}$$

where FC is fence class and NFC is non-fence class. The classified pixels are used to create the fence mask $\Omega \in R^{M\times N}$:

$$\Omega(i, j) = \begin{cases} 1 & \text{if } I(i, j) \text{ is fence pixel} \\ 0 & \text{otherwise} \end{cases}$$

During classification, some fence pixels are misclassified and some non-fence pixels are marked as fence pixels due to color similarity. Such errors can be eliminated or at least mitigated as discussed in the following section. Figure 1 shows a fence image and its respective fence mask.

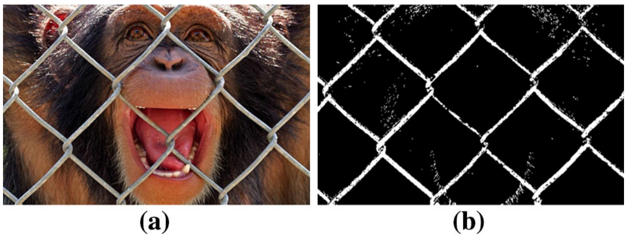

Fig. 1 Classification error. **a** An image with fence, **b** fence detected

3 Fence mask refinement

The fence detection mechanism described above may yield both *false positives*, i.e., non-fence pixels marked as fence pixels, and *false negatives*, i.e., fence pixels marked as non-fence pixels. Such misclassification generally occurs due to color variations in the fence or color similarity of non-fence pixels with the fence pixels. For accurate fence detection,

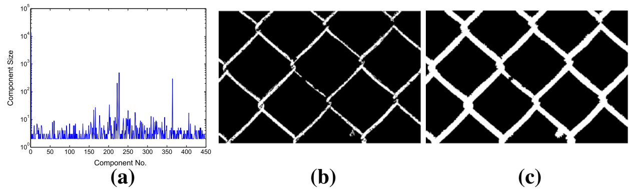

Fig. 2 **a** Size of components found in mask Fig. 1b, **b** fence mask after elimination of FP, **c** fence mask after dilation

false positives and false negatives can be reduced using morphological tools as described in the following.

False positives (FPs) usually appear as small blobs in the non-fence regions. To eliminate the FP pixels, we divide the fence mask into the corresponding connected components. The components of significant size represent the fence, whereas the small ones are usually FP pixel caused by noise or color similarity. Given the fence mask Ω, let $\{C_1, C_2, C_3, \ldots, C_n\}$ be its n connected components. The size of each component $|C_i|$ is determined and those with small size can be dropped. This goal is achieved automatically, by sorting the connected component according to their size in descending order; then, the largest ρ components are selected as parts of the fence. The number of selected components ρ is selected automatically such that:

$$\sum_{i=1}^{\rho} |C_i| \leq \alpha \sum_{j=1}^{n} |C_j| \tag{9}$$

where α specify the expected fraction of correctly classified pixels. In our experiments, it turned out that setting $\alpha = 0.95$ yields good results. Figure 2a shows the sizes of the 447 connected components found in image Fig. 1b. It can be observed that there are only few components with large size, whereas most of them contains less than 15 pixels and represent FP pixels. Figure 2b shows the fence mask after refinement.

False negatives (FNs) usually appear around the fence boundaries. These FN pixels are those characterized by a color more similar to image background than the fence model. Such pixels may also cause discontinuities in the fence mask. Such FN pixels and broken fence segments can be included in the fence mask by using the *dilation* morphological operator [26]. Dilation expands the fence regions and hence reconnects its disconnected segments. Figure 2c shows the fence mask after dilation with a typical structuring element.

4 Restoration of fence-occluded regions

The restoration of the fence region is a challenging task because the fence may extend along the entire image and may occlude a significant portion of the image, e.g., up to 40 % in some cases. In this paper, a hybrid inpainting algorithm is proposed hybrid because it combines interpolation-based and exemplar-based inpainting techniques. This algorithm is an extended version of the pyramid interpolation [11] where we added an additional exemplar-based inpainting functionality [9] to restore large regions.

The proposed inpainting algorithm works in two steps. First, the image is successively downsampled so as to reduce the area covered by the fence region; then, the fence holes are inpainted at low resolution using patch-based inpainting. Second, the inpainted region is upsampled and copied in the corresponding full-resolution image. The two steps are described in detail in the following two subsections.

4.1 Image downsampling and texture recovery

Both the image $I(m, n)$ and the fence mask $\Omega(m, n)$ are successively downsampled yielding the set of images $\{I_0, I_1, \ldots, I_S\}$, $\{\Omega_0, \Omega_1, \ldots, \Omega_S\}$, where I_k, Ω_k are the result of k downsampling by a factor 2 (on both rows and columns). The maximum downsampling level S is selected so as to reduce image sizes below a predefined threshold ϕ. Simple downsampling without anti-aliasing filter can be used. Indeed, in this case it is not critical to maintain high visual quality of the downsampled image as whole. On the contrary, we want to reduce the area covered by the fence pixels while retaining the local information of the pixels in the fence neighborhood. Thus, the width of the fence in Ω_S is considerably reduced and can be recovered more easily. We propose to recover the fence holes left in I_S using the Criminisi patch-based method. It is worth pointing out that working with a downsampled version of the image also dramatically reduces the computational cost required by inpainting.

4.2 Upsampling of the inpainted region

After inpainting, the downsampled image needs to be upsampled back to the original resolution. Let $\hat{I}_k$ represent the de-fenced image recovered at resolution level k, i.e., $\hat{I}_S$ is used to denote the inpainted image obtained at the lowest resolution. Any image $\hat{I}_k$ recovered at level k can be upsampled, e.g., using a Gaussian kernel define by:

$$\widetilde{I}_{k-1}(x, y) = \sum_{i=-2}^{i=2} \sum_{j=-2}^{j=2} w(i, j) \hat{I}_k(2x + i, 2y + j) \tag{10}$$

where $w = hh^\top$ with $h = \frac{1}{16}[1\ 4\ 6\ 4\ 1]$ is a Gaussian weighting function and $\widetilde{I}_{k-1}$ is an upsampled version of $\hat{I}_k$. Then, only the upsampled fence pixels are copied from $\widetilde{I}_{k-1}$ to the corresponding recovered image at level $k - 1$, whereas the non-fence pixels are left untouched, as follows:

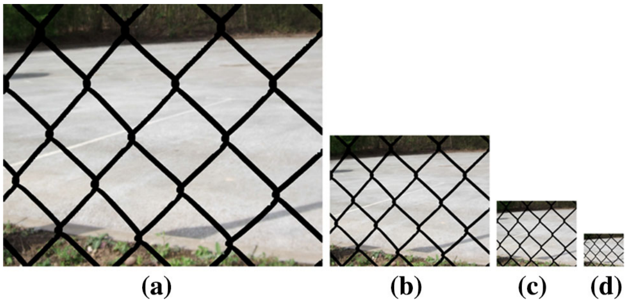

Fig. 3 **a** An image with fence mask (*in black color*), **b–d** are its three downsampled images

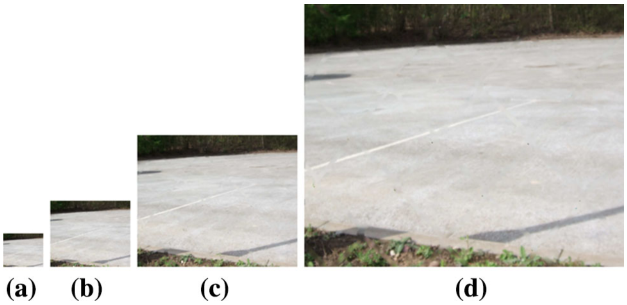

Fig. 4 Corresponding de-fenced version of images shown in Fig. 3

$$\hat{I}_{k-1}(x,y) = \begin{cases} \widetilde{I}_{k-1}(x,y) & \text{if} \quad \Omega_{k-1}(x,y) = 1; \\ I_{k-1}(x,y) & \text{if} \quad \Omega_{k-1}(x,y) = 0. \end{cases}$$

This process is iterated up to computation of $\hat{I}_0$ in which the fence region is completely recovered. Figure 3 shows an example of downsampling process with $S = 3$ ($\phi = 200$), whereas Fig. 4 shows the images obtained by inpainting and iterative upsampling. To improve the visual quality of the de-fenced image, super resolution methods, e.g., [12,13,28,33], can be used to upsample the inpainted image; however, such methods usually take considerable amount of execution time.

5 Experiments and results

In this section, the proposed image de-fencing technique is tested on several images. Performance is analyzed in terms of both fence detection accuracy and visual quality of the recovered image. We also note that the proposed algorithm can be used in a more general setting where one wishes to remove objects other than fences or even textures. Objective evaluation is carried out on the proposed hybrid inpainting algorithm using different size and types of fences. Moreover, the performance of the proposed algorithm is also compared with state-of-the-art image de-fencing and inpainting techniques. Results on 18 experiments are reported in this section, which are summarized in Table 1. The execution time has been obtained on an Intel Core i7 2.5 GHz with 8 GB RAM and includes IO time, fence detection, refinement and image recovery. The test database and the source code of the proposed technique will be released soon at the paper Web site http://www.di.unito.it/~farid/Research/defencing.html.

Table 1 Experimental details

Category	Exp.	Description	Image size	WS	SE	FOC	Time
Regular	1	Chimpanzee	509 × 332	3	7	25.73	37.08
	2	Dog	338 × 352	5	5	28.85	37.83
	3	House	800 × 600	3	7	23.90	24.91
	4	Lion	480 × 360	5	7	9.47	10.72
	5	Out door	450 × 320	3	5	32.27	52.66
	6	Road	2431 × 1996	5	9	10.92	34.18
Irregular	7	Thistle	648 × 486	5	7	7.91	11.23
	8	Sparrow	800 × 576	5	5	5.58	8.52
Occluded	9	Larkspur	340 × 269	3	7	38.55	23.9
	10	Purple	450 × 600	5	7	13.93	11.30
	11	Bear	588 × 354	3	5	20.87	60.42
	12	Snapdragon	800 × 919	5	5	12.41	9.79
General	13	Eagle	556 × 450	5	7	11.03	5.36
	14	Loggerhead	600 × 400	7	5	2.58	7.33
	15	Fence	1440 × 900	7	11	12.15	12.62
	16	Red Robin	331 × 331	5	5	24.26	16.72
	17	Garden	672 × 545	7	15	34.84	13.85
	18	Wall	972 × 648	7	5	34.34	75.79

WS is window size used in fence detection, SE means size of structuring element for dilation, FOC is percentage (%) of fence-occluded region in image. Execution time is in seconds

The values of the parameters involved in the proposed de-fencing algorithm are found empirically. The first parameter, the number of points the user must mark on the fence, depends on the variation of fence color. In case of one solid color, few points, e.g., 5, are sufficient but in case the fence color has large variation 15 to 20 points must be enough. Based on these marked points, its k neighbors are selected to increase the data samples. For thin fences, k is set to 3 and for thick fences its value may be set to 9. The size of structuring element used in Sect. 3 depends on the width of the fence; in our experiments, size ranging from 3×3 to 9×9 has been used in most cases. The value of the parameter ϕ in the the downsampling process is set to 200, which yields the best trade-off on all the images used in this work.

Figure 5 shows a series of images covered by regular fences. In these experiments, images with thin fences, thick fences and fences with different orientations are selected, and all these fences are successfully removed. The proposed fence detection algorithm can be applied to detect not only the regular but also irregular fences. Figure 6 shows two examples where irregular fences have been successfully removed. Figure 7 shows some examples where fence is occluded with foreground objects like flowers and has been detected and

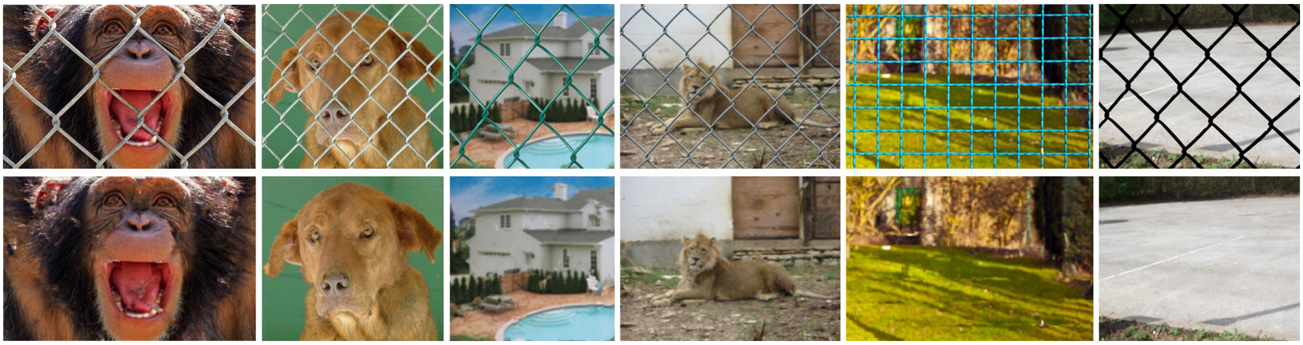

Fig. 5 Regular fence removal (Exp 1–6): original (*top*) and de-fenced (*bottom*) image

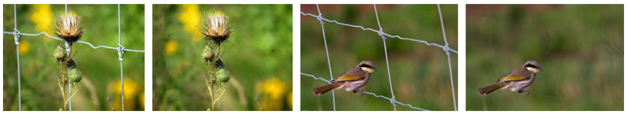

Fig. 6 Irregular fence removal (Exp. 7–8): original (*left*) and de-fenced (*right*) image

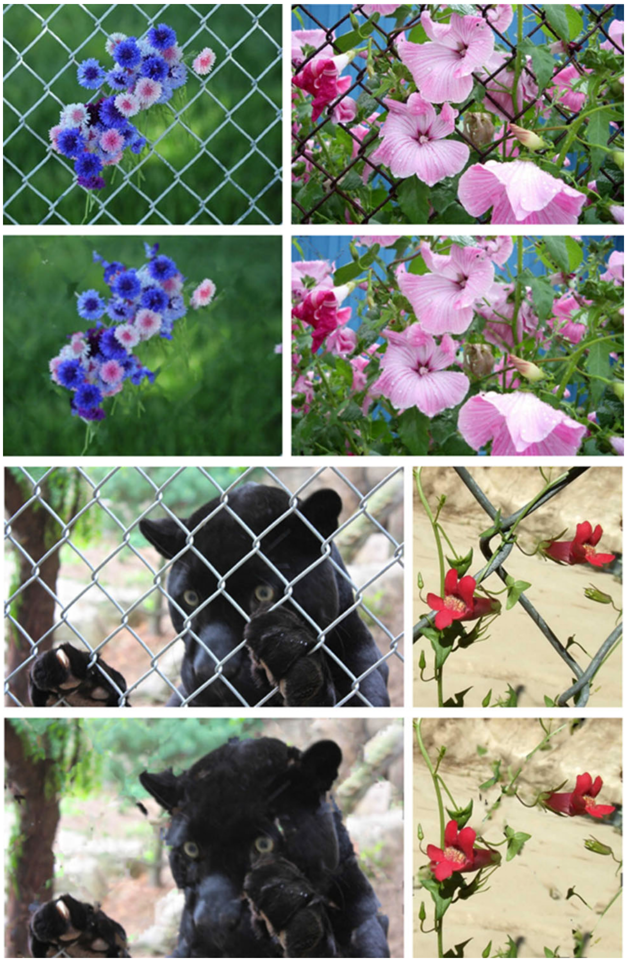

Fig. 7 Occluded fence removal (Exp. 9–12): original (*top*) and de-fenced (*bottom*) image

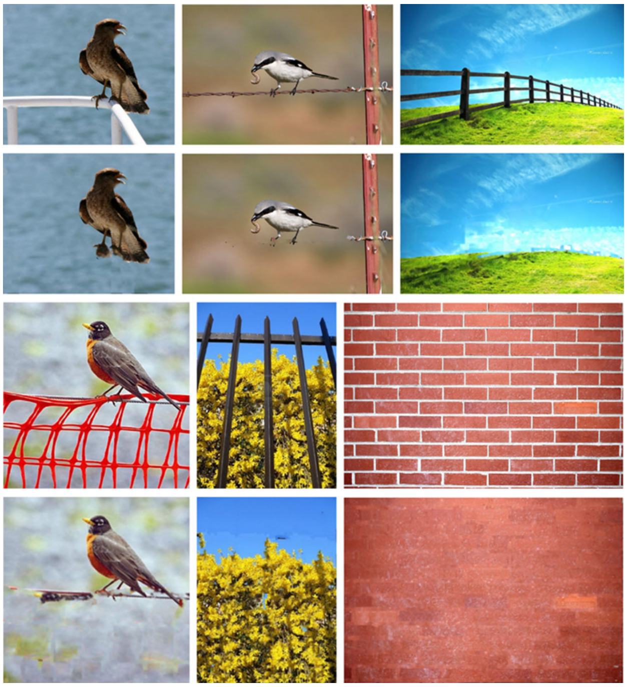

Fig. 8 *Rod*, *bars* and multiple *fence* removal (Exp: 13–18): original (*top*) and de-fenced (*bottom*) image

removed. The quality of de-fenced images demonstrates the efficiency of the proposed technique in this case as well.

The proposed algorithm can also be used for other types of occluding pattern. Figure 8 shows some examples where general objects like rods, bars and multiple fences are removed. In the last example, we applied the proposed algorithm to the case of a multicolor object, which is a fence with multiple colors. In this case, the detection algorithm can be applied iteratively, by removing fence of one color at a time. Figure 9 shows the fence colored with red, green, blue and yellow. The left image in Fig. 9 is the input image; the middle image is the result of removing the green fence. The intermediate de-fenced image is then used to train a model for blue fence removal and so on.

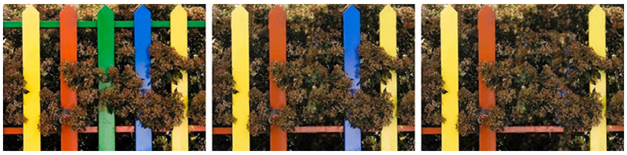

Fig. 9 Removing multiple colors: original image (*left*), *green bar* removal (*middle*), *blue bar* removal (*right*) (color figure online)

5.1 Comparison with state-of-the-art techniques

The performance of proposed algorithm is compared with three state-of-the-art image de-fencing and inpainting techniques. We compare the proposed technique with Liu et al. [17] and Park et al. [24] image de-fencing approaches to evaluate its performance in terms of fence detection and fence removal. Moreover, the proposed hybrid inpainting method is also compared with Criminisi et al. [9] inpainting algorithm in terms of image quality and execution time.

5.1.1 Comparison with image de-fencing techniques

Both Liu et al. [17] and Park et al. [24] image de-fencing approaches use lattice detection algorithm to detect the fence. After fence detection, the Liu algorithm used Criminisi et al. [9] inpainting to recover the occluded regions, whereas the Park algorithm proposed multiview inpainting when two images of the scene, captured from two very close viewpoints, are available. In case of single view, Park algorithm also relies on Criminisi et al. [9] inpainting to recover the occlusions.

As described in Sect. 1, the Liu and Park algorithms work for regular and near-regular pattern fences. Both approaches fail to detect irregular pattern fences, and in some cases they may fail to obtain a complete lattice in regular pattern fence too. On the contrary, the proposed algorithm is able to detect regular as well as irregular pattern fences (see Sect. 5). Figure 10 compares the results in terms of fence detection obtained by the proposed approach and the Liu algorithm on symmetrical fences. The symmetrical fences/objects detected by Liu algorithm are shown superimposed on the test image in yellow and red colors. These results show that Liu algorithm may fail to detect approximatively regular pattern fences, whereas the proposed algorithm successfully detects the fences in all the tested cases. Figure 11 compares the performance of the proposed algorithm and Liu approach in terms of fence removal. The figure shows two examples where the Liu algorithm successfully detects the fences but fails to recover the occluded region. The proposed approach successfully detects and removes the fences in these images. The results of Park algorithm on the previous images (Figs. 10, 11) are not available, and therefore, we could not include them in the comparison.

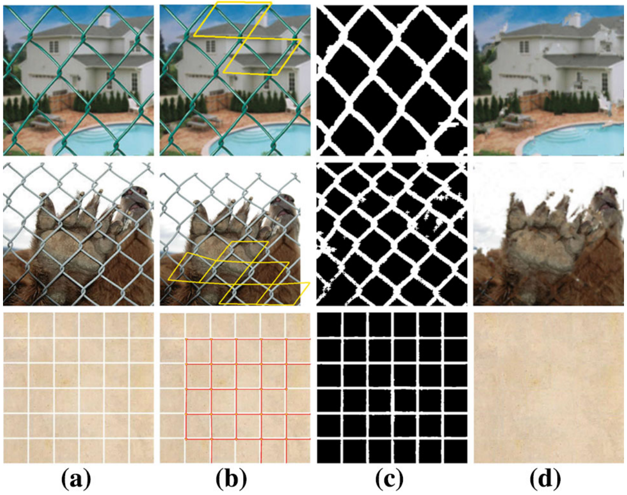

Fig. 10 Examples where Liu et al. [17] de-fencing approach fails in fence detection. **a** test image, **b** fence detected by Liu algorithm (shown in *yellow* and *red colors*), **c** fence detected by proposed algorithm, **d** de-fenced image by proposed algorithm. (These test images are are borrowed from Liu dataset: http://vision.cse.psu.edu/research/imageDe-fencing/index.shtml) (color figure online)

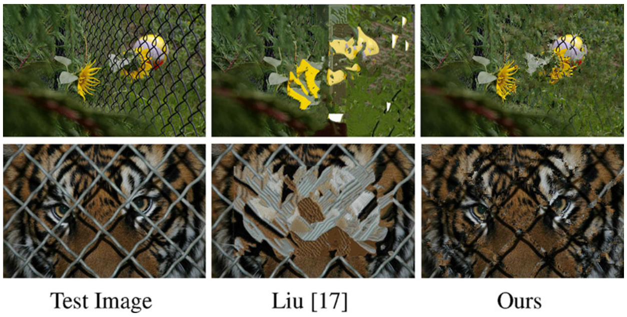

Fig. 11 Examples where Liu et al. [17] de-fencing approach fails in fence detection or in removal phase. (These test images are are borrowed from Liu dataset: http://vision.cse.psu.edu/research/imageDe-fencing/index.shtml)

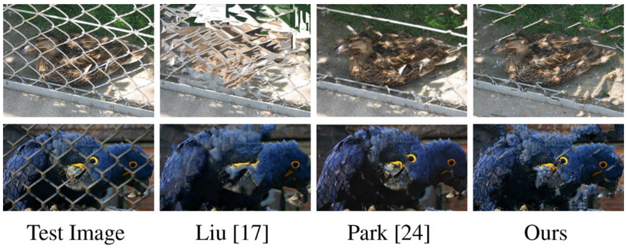

Fig. 12 Comparison with Liu and Park image de-fencing approaches

Figure 12 compares the performance of proposed algorithm with Liu and Park de-fencing approaches. Since both the proposed and the Liu algorithms are single-view image de-fencing approaches, we use single-view images in this comparison. The images used in this comparison are cour-

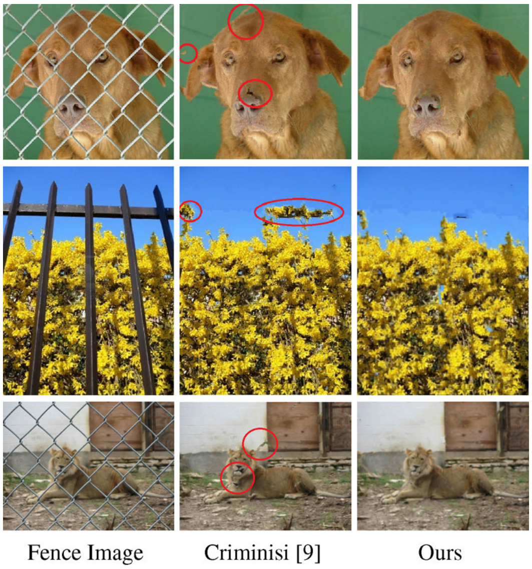

Fig. 13 Fence inpainting comparison of proposed algorithm with Criminisi et al. [9]. The main inpainting artifacts are pointed out by *red circles* (color figure online)

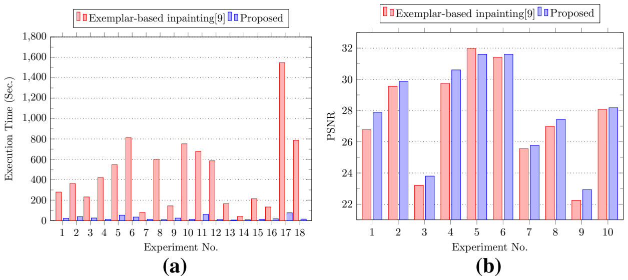

Fig. 14 **a** Execution time comparison of the proposed algorithm with [9] over 18 experiments, **b** quality comparison in terms of PSNR

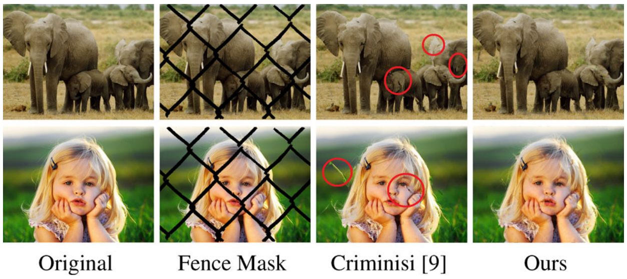

Fig. 15 Comparison of proposed hybrid inpainting with Criminisi et al. [9] on synthetic data. *Red circle* points some inpainting artifacts (color figure online)

tesy of Park et al. [23] dataset available at[1]. On such test images, it can be noted that our algorithm yields better results than Liu, whereas the Park algorithm yields slightly better image reconstruction than ours. This is mainly due to the improved lattice detection in Park algorithm. Few fence parts can still be noted in our reconstructed images since the proposed fence detection algorithm is based solely on the color model of the fence. On the contrary, the Park algorithm, in addition to color, exploits the fence structure (in the form of lattices). The results presented in Figs. 10, 11 and 12 show that the proposed algorithm performs better than Liu de-fencing approach and its performance is comparable with Park algorithm. However, in case of asymmetrical and non-uniform fences, Liu and Park algorithms do not work, whereas the proposed de-fencing algorithm is more general and can effectively remove such objects (e.g., see Figs. 6, 8).

5.1.2 Comparison with image inpainting techniques

In this section, we compare the proposed hybrid inpainting algorithm with Criminisi et al. [9] algorithm. In particular, the proposed inpainting and method in [9] are used to recover the same regions. The results show that the proposed inpainting method yields better quality, especially when the fence is large. Figure 13 shows the visual quality obtained by [9] on 3 test images. Interpolation artifacts produced by [9] can be spotted easily as compared to our results. Indeed, in the presence of large fences, the proposed technique performs better. Thanks to the multiresolution approach which decreases the fence area and produces fast and better matches for occluded regions. Moreover, the proposed hybrid inpainting technique turns to be approximately 19 times faster than exemplar-based inpainting (execution time comparison is given in Fig. 14a). This gain in speed is again due to our multiresolution approach.

Finally, we compare the obtained image quality versus [9] also in terms of peak signal-to-noise ratio (PSNR). These experiments are carried out with synthetic data where we manually occlude some regions with a known fence mask, thus allowing us to compute PSNR of the recovered image with respect to the original one. Ten experiments are performed on five test images (624 × 480) with two fences with different widths. The results are presented in Fig. 14b, which shows that the proposed technique is consistently better in restoring the fence region with an average gain of 0.5 dB. Figure 15 shows two examples of de-fencing on synthetic data set. The results demonstrate that the proposed algorithm is better than existing algorithms both in terms of PSNR and visual quality.

6 Conclusion

A novel approach for removing fences from digital photographs is presented in this paper. The proposed technique with little user assistance constructs a fence color model, and a two-level classification algorithm segments the fence

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

from the background. The detected fence is refined through connected component analysis and by using morphological operators. To recover the fence region, a hybrid inpainting technique is proposed that integrates the exemplar-based inpainting with pixel interpolation method based on pyramid. The proposed de-fencing algorithm is capable to detect symmetrical as well as asymmetrical fences and restores the fence regions efficiently. Moreover, the experiments point out that the proposed technique can be generalized to the problem of removing other kind of unwanted distributed patterns.

References

1. Arias, P., Facciolo, G., Caselles, V., Sapiro, G.: A variational framework for exemplar-based image inpainting. Int. J. Comput. Vis. **93**, 319–347 (2011)
2. Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G., Verdera, J.: Filling-in by joint interpolation of vector fields and gray levels. IEEE Trans. Image Process. **10**(8), 1200–1211 (2001)
3. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: SIGGRAPH, pp. 417–424 (2000)
4. Bertalmio, M., Vese, L., Sapiro, G., Osher, S.: Simultaneous structure and texture image inpainting. IEEE Trans. Image Process. **12**(8), 882–889 (2003)
5. Bornemann, F., März, T.: Fast image inpainting based on coherence transport. J. Math. Imaging Vis. **28**, 259–278 (2007)
6. Bruni, V., et al.: Semi-transparent blotches removal from sepia images exploiting visibility laws. Signal Image Video Process. **7**(1), 11–26 (2013)
7. Cheng, W.H., et al.: Robust algorithm for exemplar-based image inpainting. In: Proceeding of International Conference on Computer Graphics, Imaging and Visualization, pp. 64–69 (2005)
8. Criminisi, A., Perez, P., Toyama, K.: Object removal by exemplar-based inpainting. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 721–728 (2003)
9. Criminisi, A., Perez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Process. **13**(9), 1200–1212 (2004)
10. Farid, M., Khan, H.: Image inpainting using dynamic weighted kernels. In: Proceeding of the IEEE International Conference on Computer Science and Information Technology (ICCSIT), vol. 8, pp. 252–255 (2010)
11. Farid, M., Khan, H., Mahmood, A.: Image inpainting based on pyramids. In: Proceeding of the IEEE International Conference on Signal Processing (ICSP), pp. 711–715 (2010)
12. Freeman, W., Jones, T., Pasztor, E.: Example-based super-resolution. IEEE Comput. Graph. Appl. **22**(2), 56–65 (2002)
13. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: Proceedings of the IEEE ICCV, pp. 349–356 (2009)
14. Hays, J., Leordeanu, M., Efros, A.A., Liu, Y.: Discovering texture regularity as a higher-order correspondence problem. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 522–535 (2006)
15. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1482–1489 (2005)
16. Leung, T.K., Malik, J.: Detecting, localizing and grouping repeated scene elements from an image. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 546–555. Springer (1996)
17. Liu, Y., Belkina, T., Hays, J., Lublinerman, R.: Image de-fencing. In: Proceedings of the IEEE CVPR, pp. 1–8 (2008)
18. Liu, Y., Collins, R., Tsin, Y.: A computational model for periodic pattern perception based on frieze and wallpaper groups. IEEE Trans. Pattern Anal. Mach. Intell. **26**(3), 354–371 (2004)
19. Lobay, A., Forsyth, D.A.: Shape from texture without boundaries. Int. J. Comput. Vis. **67**, 71–91 (2006)
20. Mirkamali, S., Nagabhushan, P.: Object removal by depth-wise image inpainting. Signal Image Video Process. **9**(8), 1785–1794 (2015)
21. Oliveira, M.M., Bowen, B., Mckenna, R., sung Chang, Y.: Fast digital image inpainting. In: Proceedings of the International Conference Visualization, Imaging and Image Processing (VIIP), pp. 106–107 (2001)
22. Papafitsoros, K., Schnlieb, C.: A combined first and second order variational approach for image reconstruction. J. Math. Imaging Vis. **48**(2), 308–338 (2014)
23. Park, M., et al.: Deformed lattice detection in real-world images using mean-shift belief propagation. IEEE Trans. Pattern Anal. Mach. Intell. **31**(10), 1804–1816 (2009)
24. Park, M., Brocklehurst, K., Collins, R.T., Liu, Y.: Image de-fencing revisited. In: Proceedings of the Asian Conference on Computer Vision (ACCV), pp. 422–434 (2011)
25. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990)
26. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press Inc., Orlando (1983)
27. Shen, J., Chan, T.F.: Mathematical models for local nontexture inpaintings. SIAM J. Appl. Math. **62**(3), 1019–1043 (2002)
28. Tian, J., Ma, K.K.: A survey on super-resolution imaging. Signal Image Video Process. **5**(3), 329–342 (2011)
29. Tuytelaars, T., Turina, A., Van Gool, L.: Noncombinatorial detection of regular repetitions under perspective skew. IEEE Trans. Pattern Anal. Mach. Intell. **25**, 418–432 (2003)
30. Wu, J., Ruan, Q.: Object removal by cross isophotes exemplar-based inpainting. Proc. Int. Conf. Pattern Recognit. (ICPR) **3**, 810–813 (2006)
31. Xu, Z., Sun, J.: Image inpainting by patch propagation using patch sparsity. IEEE Trans. Image Process. **19**, 1153–1165 (2010)
32. Yang, C.K., Yeh, Y.C.: Stain removal in 2D images with globally varying textures. Signal Image Video Process. **8**(7), 1373–1382 (2014)
33. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution via sparse representation. IEEE Trans. Image Process. **19**(11), 2861–2873 (2010)