Signal, Image and Video Processing
https://doi.org/10.1007/s11760-018-1266-0

ORIGINAL PAPER

Image de-fencing using histograms of oriented gradients

Madiha Khalid[1] · **Muhammad Murtaza Yousaf**[1] · **Kashif Murtaza**[1] · **Syed Mansoor Sarwar**[1]

Received: 20 September 2016 / Revised: 1 March 2018 / Accepted: 2 March 2018

Abstract
Image de-fencing is often used by digital photographers to remove regular or near-regular fence-like patterns from an image. The goal of image de-fencing is to remove a fence object from an image in such a seamless way that it appears as if the fence never existed in the image. This task is mainly challenging due to a wide range intra-class variation of fence, complexity of background, and common occlusions. We present a novel image de-fencing technique to automatically detect fences of regular and irregular patterns in an image. We use a data-driven approach that detects a fence using encoded images as feature descriptors. We use a variant of the histograms of oriented gradients (HOG) descriptor for feature representation. We modify the conventional HOG descriptor to represent each pixel rather than representing a full patch. We evaluated our algorithm on 41 different images obtained from various sources on the Internet based on a well-defined selection criteria. Our evaluation shows that the proposed algorithm is capable of detecting a fence object in a given image with more than 98% accuracy and 87% precision.

Keywords Image de-fencing · Histograms of oriented gradients · Supervised learning and classification · Object detection · Image inpainting · Fence removal

1 Introduction

Modifying images in an undetectable form is not a new art; it is as old as artistic creation itself. Digital inpainting has provided numerous benefits to this art. Bertalmio et al. [1] first introduced the term digital inpainting. The applications of digital inpainting include restoration of damaged paintings, removal of cracks and scratches from photographs, and elimination of selected objects like fences and wires, superimposed objects like stamps, dates, and publicity statements. The task of removing fence-like foreground patterns from an image is called image de-fencing [2]. Image de-fencing is a real-life problem of image inpainting, where fences from images are removed in a visually plausible way. Often such fence objects are unwanted but unavoidable in photography. A few examples of images with fences include pictures of animals in cages, photographs taken through wired windows, and photographs of children playing across a railing. Such images need de-fencing in order to improve their esthetic quality for commercial as well as personal photography. There are several challenges in fence detection due to many variations in the appearance of fences, including their color, shape, and geometry. Secondly, the background of the target image can be wide-ranging and complex. Finally, common illumination and occlusions create further obscurity in images. Thus, a robust fence detector should be able to single out a fence object from a complex background regardless of the variations between the fence and background image, illuminations, and partial occlusions.

We propose a generalized image de-fencing framework that can work well for both regular and irregular fence patterns. The framework automatically detects the fence in an image and removes it in a seamless way. As part of the framework, a generalized feature-based algorithm works for automatic fence detection based on supervised learning and classification. A classifier that works on the basis of edge directions and distribution of intensity gradients is trained to classify fence pixels and non-fence pixels. Since the classifier uses no prior information about the type of fence in the image, therefore, no manual marking is required on the image. The whole process of fence detection and fence removal is com-

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s11760-018-1266-0) contains supplementary material, which is available to authorized users.

✉ Madiha Khalid
madiha.khalid@pucit.edu.pk

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

Published online: 12 March 2018

pletely automatic. To the best of our knowledge, the proposed approach is the only one that is fully automatic and works for both types of fence structures. We carried out a series of carefully designed experiments to test our algorithm for quality and performance by visual inspection as well as through quantitative measures. We also compare our algorithm with state-of-the-art image de-fencing algorithms.

The term image de-fencing was first introduced by Liu et al. [2]. They use a three-phase de-fencing approach in which first a potential fence structure is detected as de-formed lattice using image segmentation algorithm [3]. The background and foreground layers are separated, and background is filled using a patch-based inpainting algorithm. Improvement in this work is discussed in [4], and the improved algorithm not only considers texels but also computes a two-dimensional ($2D$) vector. Both the algorithms are limited to regular or near-regular fences only. Farid et al. [5] proposed a semiautomatic algorithm that identifies fence region by estimating color models. Zou et al. [6] exploit depth and color information to remove fence structure from RGBD images. The work described in [7] uses signal de-mixing technique to obtain sparse and periodic properties of a near-regular fence. Another study [8] uses a parallel algorithm for image de-fencing. The research described in [9] uses spectral information to distinguish fence from other objects.

In spite of much recent research on image de-fencing [2,4–7,9], completely automatic fence detection is still an open problem. However, video de-fencing can lead to relatively better results due to the availability of missing information in the neighboring frames. A number of recent studies [10–19] have considered the problem of image de-fencing when multiple images of the same scene are captured by moving a camera or a video clip is available. For example, Xue et al. [11] propose an algorithm that uses a sequence of images taken by slightly moving the camera. The algorithm uses pixel-wise flow field motion representation and edge flow method to separate the foreground mask and the background image. Mu et al. [12] use visual parallax to segment the fence from background. Although both [11] and [12] successfully de-fence a video, however, they are limited to static scenes only. Khasare et al. [13] use interactive labeling of fence pixels. Negi et al. [14] propose a framework to simultaneously improve the quality of a low- resolution video and eliminate fence pixels from the occluded frames. Their algorithm exploits the subpixel displacement between low-resolution video frames to retrieve a high-resolution fence-free video. Some techniques [9,20,21] detect the fence from multifocus images using a set of images (with different focal lengths, or flashlight on or off) of the same scene as input. Yi et al. [15] use color and motion features to cluster pixels and automatically detect the fence from dynamic videos using graph cut optimization. Research described in [16,17] uses convolutional neural networks to segment out the fence from a single video frame and exploits optical flow algorithm to find correspondence between the reference frame and other frames of the video. Another work [18] suggests a multimodal approach that uses aligned depth maps of multiple frames of a video obtained through the Microsoft Kinect sensor. It has been observed that all the image de-fencing techniques discussed thus far either use color models, assume a particular geometry, involve user intervention to take initial clues about the fence, or require multiple images of a scene. However, the proposed algorithm is completely automatic and can detect fences regardless of the color, geometry, and orientation of the fence in an image. Its success mainly comes from the fact that our algorithm does not assume any particular fence color, geometry, or structure.

Since image de-fencing is a two-way problem, i.e., fence detection and fence removal, where latter is typically achieved by an inpainting algorithm. Considerable literature is available on image inpainting [1,22–28]. However, most of the work on image de-fencing uses already established image inpainting algorithms, or a variant of them, to seamlessly remove the fence and fill in the holes created by its removal. For example, Liu et al. [2] and Yi et al. [15] use texture-based inpainting proposed by Criminsi et al. [22] to fill in the void created by the fence region. Similarly, the work presented in [6] also adapts the exemplar-based method to fill in the fence region. The image de-fencing algorithm reported in [7] adopts a texture-based inpainting technique, COMEP, proposed in [26]. The algorithm proposed in [5] uses a hybrid inpainting technique that is a combination of the exemplar-based [22] and pyramid-based [29] inpainting. Park et al. [4] use multiview inpainting for image restoration. Several video de-fencing techniques [11–14,17] inpaint the fence region by retrieving the missing information from adjacent frames of the video. The work described in [9] restores the occluded region by using approximate nearest neighbor search algorithm.

The rest of the paper is organized as follows: Sect. 2 details the proposed framework. Section 3 presents experimental results with quality assessment. Finally, the paper concludes with future directions for further research on the topic in Sect. 4.

2 The approach

The overall architecture of the proposed approach is divided into three major phases: training/learning, fence detection, and restoration. The most important phase is training a classifier for fence/non-fence classification. The training data are generated from several images by labeling pixels from both fence and non-fence classes. A binary classifier is trained by feeding it the feature sets of the generated training data. The fence detection phase identifies the fence based on a

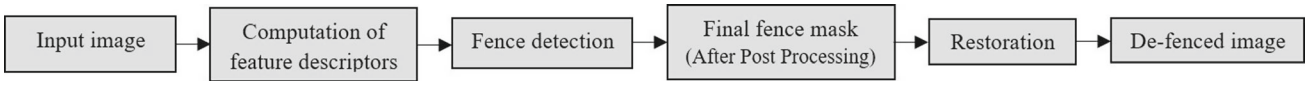

Fig. 1 An overview of proposed image de-fencing algorithm

feature set (i.e., the feature vector) that is computed for each pixel. This feature vector is rooted on the computation of histograms of oriented gradients (HOG). Then, the trained binary classifier classifies the pixels as fence or non-fence. After getting a fence mask from the classifier, we refine fence mask using morphological operators in order to connect the broken boundaries of fence.

As our work mainly focuses on fence detection, we used a reliable existing restoration technique [22] that eliminates fence region by using exemplar-based texture synthesis. Figure 1 summarizes the flow of proposed image de-fencing approach. In the following subsection, we will discuss the feature set and its computation.

2.1 The feature set

Descriptors are feature vectors that are used to distinguish different patterns and trends of the images. HOG [30] descriptors are used to detect objects in digital images. It is analogous to SIFT descriptors [31], shape contexts [32], and edge orientation histograms [33]. It performs computations on dense and overlapping grids of connected blocks in an image.

The HOG performs computation on two levels of image regions: cells and blocks. Cells are small spatially connected fixed sized regions of images, while blocks are larger connected regions of images. The algorithm first divides images into cells and then computes histograms of gradient directions for each pixel within each cell. The combination of these histograms forms the descriptor. To minimize error, histograms are normalized by computing the intensity value on each block and using the resultant value to normalize all of the cells in respective blocks. These normalized blocks are referred to as normalized HOG descriptors. Each step of the HOG computation is briefly explained bellow:

Step 1 Color normalization This step applies an optional color normalization. It can be skipped because normalizing color and gamma values offers minimal impact on performance [30]. The reason of limited impact of normalization in HOG computations is that the successive descriptor normalization exhibits the same results. Thus, initial color normalization and gamma equalization can be passed over, leaving no impact on results.

Step 2 Gradient computation This step computes the first-order image gradients. A one-dimensional $(1D)$ discrete derivative mask in both horizontal and vertical directions or a two- dimensional $(2D)$ discrete derivative mask is applied to compute a gradient. For a given image I, the gradient magnitude is computed as

$$\|\mathrm{GM}\| = \sqrt{{I_x}^2 + {I_y}^2} \tag{1}$$

where I_x and I_y are obtained as follows:

$$I_x = I \otimes X, \qquad I_y = I \otimes Y \tag{2}$$

Step 3 Orientation binning In this step, cell histograms are computed on the basis of gradient direction. Gradient direction is defined as:

$$\theta = a\tan(I_y, I_x) \tag{3}$$

where θ is gradient direction, I_y is first- order derivative in y direction and I_x is first-order derivative in x direction as computed in Eq. 2. Each pixel donates a weighted vote in favor of an orientation histogram based on its own gradient direction. These votes are accumulated in orientation bins within each cell. Cells are small physically connected regions of an image. The vote is computed at each pixel on the basis of its gradient magnitude. It can be the square root of gradient magnitude, squared gradient value, simply the magnitude, or a function of the magnitude in any form.

Step 4 Normalizing descriptor blocks Since gradients highly fluctuate in background/foreground contrasts and illuminations, therefore, local normalization of gradients is required in order to facilitate these variations. To normalize gradients, cells are clustered into larger, spatially coupled regions called blocks and each block is separately normalized. These blocks are overlapping in computations so that each cell can potentially contribute to the final descriptor vector multiple times. The blocks are of two types according to their geometries, rectangular and circular. These block descriptors are normalized as:

$$\upsilon \leftarrow \upsilon/(\|\upsilon\|_1 + \varepsilon) \tag{4}$$

where υ is the un-normalized descriptor vector, $\|\upsilon\|_1$ is 1-norm and is a normalization constant introduced to keep away from divide-by-zero error, and ε is a small normalization constant. We referred to normalized block descriptors as our final HOG descriptors that are ready to be fed to some supervised learning-based identification system for training.

2.2 Training/learning

The fundamental step in training is the training data and its formation. To generate training data, take a collection of

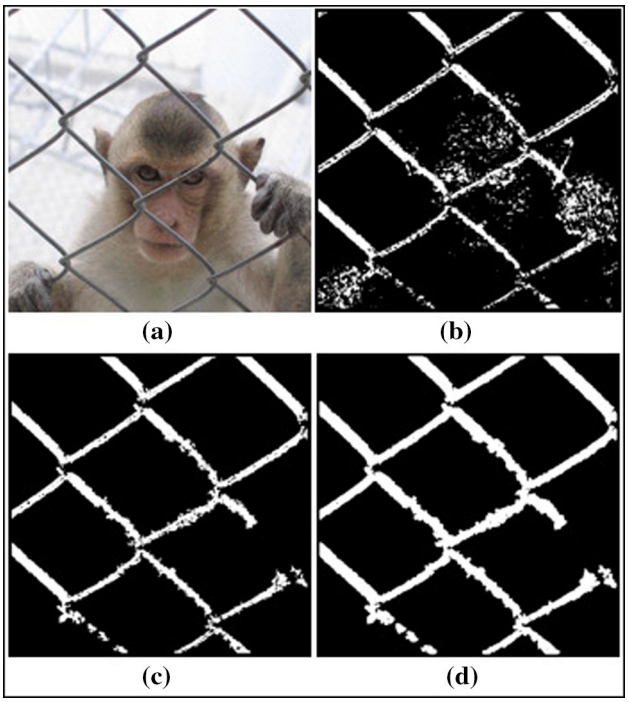

Fig. 2 Fence detection and postprocessing. **a** Original image, **b** fence detected using the proposed approach, **c** the results after eliminating false positives from the fence mask, **d** final mask after eliminating false negatives

images such that each image in the collection is a true representative of fenced image and for each image label sample pixels from both classes, i.e., fence class and non-fence class. Then, compute HOG descriptors of labeled pixels. Train a classifier by feeding it marked pixels along with their descriptors.

2.3 Fence detection

We compute the HOG vector for each pixel of the target image and project that vector on to the space spanned by the bases vectors obtained by reducing the number of variables and pass this structure to the trained classifier. The classifier comes up with a binary fence mask. Figure 2 shows an example fence image and the obtained fence mask. As shown in Fig. 2b, false positives and false negatives may arise in obtained fence mask. To eliminate false positives and negatives, some postprocessing is required that is explained in the next section.

2.4 Postprocessing

The fence mask generated by the binary classifier may contain some non-fence pixels masked as fence pixels. These false positives are eliminated by computing connected components, discarding the smaller size connected components, and retaining the larger connected components, and we eliminate the false positives, as described in [5]. Figure 2c shows the refined fence mask after eliminating false positives. There may be some pixels between the fence pixels that are marked as non-fence and are not included in the fence mask. These false negatives are handled by an elementary morphological operator, called Dilation [34].

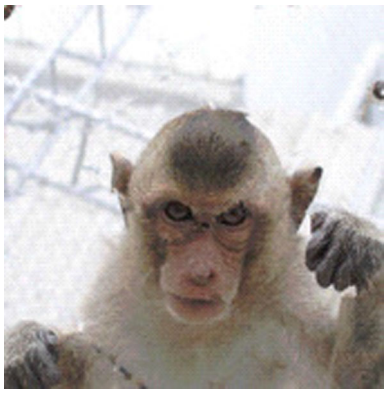

Fig. 3 A restored image

2.5 Restoration

After successful fence detection, we obtain a binary fence mask. The next step is to plausibly remove the fence masked pixels from the image. For this purposed, we relied on Criminisi et al. [22] method for region filling that carefully propagates structure and texture information to fill in the target regions. This method uses an exemplar-based texture synthesis technique that determines the order of filling in the marked region. Given the obtained fence mask and original image, the algorithm [22] can produce a fence-free image. A restored image using this algorithm is shown in Fig. 3.

3 Evaluation

A series of experiments was performed for quality assessment of the proposed algorithm. This section discusses the experimental setup, and the results obtained during the experimentation.

3.1 Dataset

For the evaluation of proposed algorithm, thirty-five different images were taken for training dataset from the Internet. These images were of varying size and resolution. As our classifier is purely data driven and the performance of the proposed algorithm is directly reliant on the training of the classifier, so we ensured that the training dataset contains maximum variants in type and structure of the fence. We

empirically found that in order to generate training data, it is sufficient to label twenty or more fence and non-fence pixels in each image; thus, the same was used to generate training data. After training the classifier on training dataset, the proposed algorithm was tested on a variety of images. The test dataset consists of forty-one distinct images obtained from the Internet. For a fair evaluation, we kept the two image collections (i.e., images in test dataset and images in train dataset) disjoint. To assess the performance of our algorithm on a broad spectrum of fences, we ensured that our test dataset contains a variety of fence structures in terms of texture, color, shape, and geometry. Furthermore, images having fences with partial occlusions and common illumination were also included in test dataset. Out of forty-one images in our test dataset, eighteen images contained regular fence structures, thirteen contained regular but partially occluded or obstructed fences, and ten images contained irregular or multistructure fences.

3.2 Results and discussion

The parameter values involved during feature computation and post processing were found empirically. For gradient computation, larger and complex masks result in poor performance while the simpler ones give best results [35], so we used the following 1D centered discrete mask in both horizontal (X) and vertical (Y) directions with no smoothing effects:

$$X = \begin{bmatrix} -1, & 0, & 1 \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} -1, & 0, & 1 \end{bmatrix}^{\mathrm{T}}$$

For binning the orientation histograms, we used the gradient magnitude itself as weighted vote because the bare magnitude value provides the finest outcome, while other forms cause a notable decline in the performance graph. We used R-HOG in our experiments with parameter values defined in Table 1 that resulted in 144 features against each pixel. To establish the effect of the dilation on the fence structure, we used the square structuring element of size 5×5. For classification, we used the support vector machine (SVM) classifier. SVM proved to be simplest and reliable classifier during our initial experiments, but it took longer to converge on very large datasets. Thus, we reduced the dimensions of our feature vectors using PCA and reduced vectors were used to train the SVM.

Figure 4 shows some of the promising results of images having regular fence structures with the outcome of every intermediate stage. Figure 4a shows three images with regular fence structures, Fig. 4b shows the obtained fence mask by the proposed fence detector, and Fig. 4c, d demonstrates refined fence mask and restored image, respectively. Figure 5 shows the results of proposed algorithm for irregular fences. It can be seen that the image in first example has the fence with same color as the background object and the proposed algorithm has successfully figured the fence from background object and de-fenced the image. The second example in Fig. 5 has two geometries of fence, i.e., line shaped and diamond shaped, and the proposed algorithm has successfully detected and removed both the structures from the image. We have also evaluated our algorithm on the images with occluded fence. Figure 6 shows the images with partial occlusions and their de-fenced results generated by our algorithm. In the first image, the fence is occluded with leaves due to which a part of the fence topology has changed its regular geometry. Similarly, the other two images in the set have occluded fence regions that cause a serious alteration in the uniform structure of the fence; however,

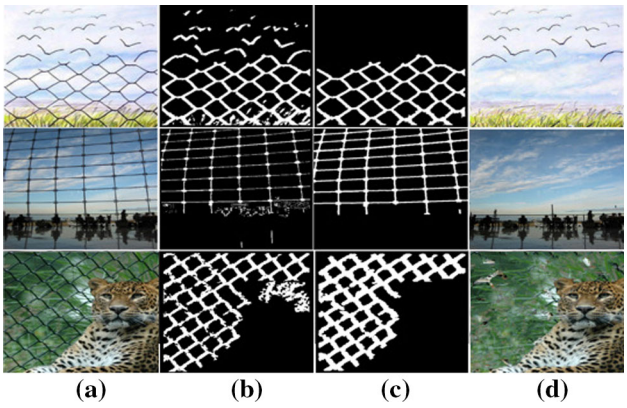

Fig. 4 Some examples of regular fence removal from images. **a** Original image, **b** detected fence mask, **c** refined fence mask, **d** restored image

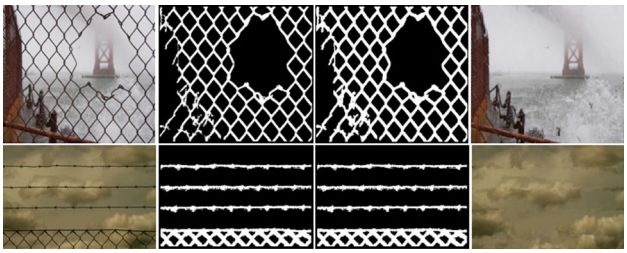

Fig. 5 Two examples of irregular fence removal using proposed algorithm. (from left to right) Original image, detected fence mask, refined fence mask, and restored image

Table 1 Parameter values used to compute HOG

Cell size	Cell overlap	Bins	Block size	Block overlap	Patch size	Patch step
[2 2]	[1 1]	9	[2 2]	[0 0]	[10 10]	[1 1]

Fig. 6 Results of images with occluded fence structures. (from left to right) Original image, detected fence mask, refined fence mask, and restored image

the proposed approach has efficiently detected the occluded fence and restored it.

3.3 Quantitative analysis

In order to evaluate the accuracy of our algorithm, we used error estimation. For this purpose, we used benchmarked images. The ground truth was generated by manually marking the fence structure. After manual marking of a fence, the actual fence mask is generated. Then, the error between the actual fence mask and fence mask obtained through our fence detection algorithm is calculated in order to measure the accuracy of our classifier. We computed two types of errors: false positive (FP) and false negative (FN). We reported the results of nine images for error estimation. These results are presented in Table 2. These results indicate a variation between the percentage of FPs and FNs. As can be seen from Table 2, the percentage of FPs is relatively higher than the percentage of FNs. The reason being that the algorithm applies dilation in postprocessing that thickens the obtained fence mask and as a result the overall count of FPs is increased. On the other hand, we achieved a good accuracy level for FNs. For assessing the overall performance of our fence detection algorithm, we used the following statistical measures: sensitivity, precision, accuracy, and specificity. These results are presented in Table 3. Analysis of these statistics reveals that the proposed approach achieved more than 98% accuracy and 87% precision. The analysis of accuracy and precision has proved that the proposed technique gives a significantly low deviation of measurement from the bench marked data. Moreover, the values of sensitivity and specificity reflect the performance of fence detector. It is evident from the results that our fence detector detects fences with significant reliability (i.e., true-positive rate 97% and true negative rate

Table 2 Error estimation of the proposed technique

Image	Size	Ours		Liu [2]		Farid [5]	
		FNs	FPs	FNs	FPs	FNs	FPs
Seaport	640 × 425	2.1	14.0	9.5	6.1	0.3	27.1
Pigeon	1600 × 1067	1.7	10.2	8.6	4.0	1.2	7.9
Sky	345 × 375	0.7	10.4	8.4	3.9	9.5	35.4
Lion	224 × 263	1.6	16.3	6.6	6.4	5.0	23.6
Monkey	452 × 446	3.3	4.9	9.7	0.9	0.4	14.0
City	246 × 255	4.0	21.0	8.1	15.1	15.2	17.7
Sparrow	259 × 194	1.4	8.3	10.2	2.4	8.6	15.2
Face	275 × 183	2.6	10.5	11.8	0.1	1.4	31.4
Clouds	400 × 424	1.5	9.3	8.6	1.3	0.5	73.4

Results of 9 images with their sizes in pixels and percentages of false positives (FPs) and false negatives (FNs)

Table 3 Overall performance of the proposed technique

Sensitivity	Precision	Accuracy	Specificity
97.56%	87.85%	98.77%	87.98%

87%). Furthermore, a separate set of experiments has been performed to evaluate the performance of proposed algorithm versus [2] and [5]. The test images for these experiments were prepared by superimposing fence on top of regular images. The de-fenced images are compared to original images using peak signal-to-noise ratio (PSNR). From Table 4, it can be seen that the obtained PSNR values represent satisfactory reconstruction of target images.

3.4 Comparison with state-of-the-art techniques

In comparison with existing image de-fencing techniques, our proposed algorithm for fence detection outperforms other techniques in terms of generality and requires no user intervention. Algorithms described by Liu et al. [2] and Park at el. [4] use lattice detection algorithm to detect fence and, therefore, work for regular or near-regular pattern fences only. On the other hand, our algorithm has the ability to detect regular, irregular, and near- regular fences with greater accuracy. Figure 7 compares the fence detection results of Liu et al. [2] and Farid et al. [5] with our algorithm.[1] The obtained fence mask is highlighted in red. It can be seen from Fig. 7 that Liu's algorithm failed to obtain a complete fence structure even in case of some regular fences. Image sequence in Fig. 8 provides an example that contains two geometries of the fence structure. The Liu algorithm partially detected the diamond shaped fence only, whereas our algorithm suc-

[1] These results are produced by using the source code available at: http://vision.cse.psu.edu/data/PAMI09Win7Matlab201264bit and http://www.di.unito.it/~farid/Research/defencing.html, respectively.

Table 4 Quantitative evaluation using PSNR (in dB)

Exp.	Image	Size	Our	Liu [2]	Farid [5]
1	Highway	960 × 639	33.54	19.45	27.98
2	Beach	960 × 637	26.53	15.16	16.47
3	Rabbit	806 × 454	30.95	13.18	31.18
4	Wolf	799 × 591	26.73	11.37	27.85
5	Seagull	799 × 599	22.64	16.19	26.63

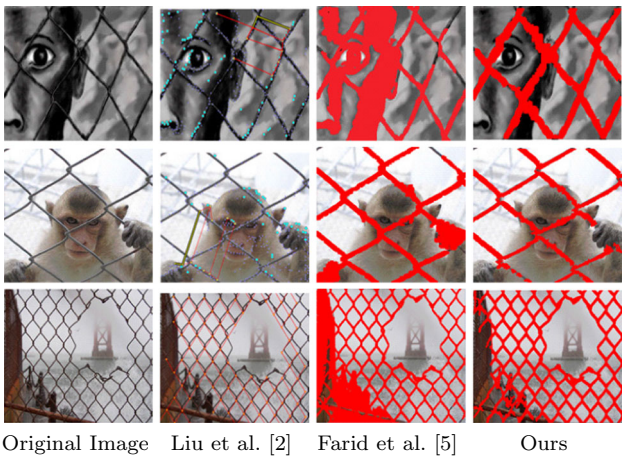

Fig. 7 Fence detection comparison of proposed algorithm with Liu et al. [2] and Farid et al. [5]

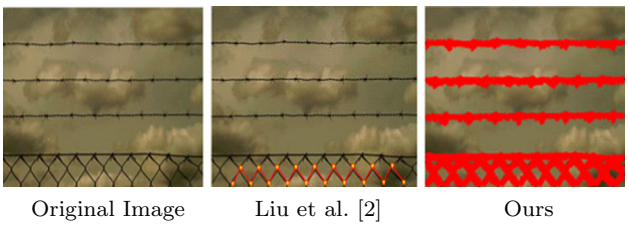

Fig. 8 Comparison of fence detection of proposed algorithm with Liu et al. [2] algorithm

cessfully detected both types of fences in the image. These results show that Liu's algorithm exhibits erroneous detection in case of irregular fences, and sometimes, even in case of regular fences. Also, Liu's algorithm can only detect quadrilateral shaped fences, whereas our algorithm outperforms Liu et al. [2] in several cases.

Farid et al. [5] fence detection algorithm is solely based on the color models of the fence, which makes it work well only in case the fence color is not present in any of the background or foreground objects. As shown in Fig. 7, Farid's fence detection algorithm obtains several non-fence parts of the image that have similar color as fence. Moreover, the algorithm in [5] requires manual point marking of fence and non-fence pixels. In contrast, our algorithm is completely automatic and exploits structural features and, therefore, shows convincing results for all types of fences.

4 Conclusion and future work

We introduce a novel approach to image de-fencing based on supervised learning. Our main contribution is the development of a fence detector that characterizes fence pixels based on a classifier. Our classifier uses feature encoding of the target image and performs fence/non-fence classification. The approach does not involve any thresholding and does not depend on the shape or color intensities of the fence. Furthermore, it does not involve any type of user intervention, and automatically detects and restores the fence of any geometry and color.

The algorithm was tested on a variety of images. Our evaluation shows promising performance of our technique for regular, irregular, and occluded fence structures. A challenging task for future research is to introduce online learning for the classifier. Thus, instead of one time training, the classifier can train itself after each incorrect classification. Another promising line of research is the investigation of the state-of-the-art inpainting techniques to find out the most suitable inpainting technique for image de-fencing. Such a study can lead to better visual results.

References

1. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: SIGGRAPH, pp. 417–424 (2000)
2. Liu, Y., Belkina, T., Hays, J.H., Lublinerman, R.: Image de-fencing. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2008)
3. Hays, J., Leordeanu, M., Efros, A.A., Liu, Y.: Discovering texture regularity as a higher-order correspondence problem. In: Proceedings of European Conference on Computer Vision (ECCV), pp. 522–535 (2006)
4. Park, M., Brocklehurst, K., Collins, R.T., Liu, Y.: Image de-fencing revisited. In: Proceedings of Asian Conference on Computer Vision (ACCV), pp. 422–434 (2011)
5. Farid, M.S., Mahmood, A., Grangetto, M.: Image de-fencing framework with hybrid inpainting algorithm. Signal Image Video Process. **10**(7), 1193–1201 (2016)
6. Qin Zou, Y., Cao, Q.L., Mao, Q., Wang, S.: Automatic inpainting by removing fence-like structures in rgbd images. Mach. Vis. Appl. **25**(7), 1841–1858 (2014)
7. Kumar, V., Mukherjee, J., Mandal, S.K.D.: Image defencing via signal demixing. In: Proceedings of 10th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP), pp. 1–8 (2016)
8. Khalid, M., Yousaf, M.M.: Parallel image de-fencing: technique, analysis and performance evaluation. In: Advanced Computer and Communication Engineering Technology, pp. 979–988. Springer (2016)
9. Zhang, Q., Yuan, Y., Lu, X.: Image de-fencing with hyperspectral camera. In Proceedings of the International Conference on Computer, Information and Telecommunication Systems (CITS), pp. 1–5 (2016)
10. Jonna, S., Satapathy, S., Sahay, R.R.: Stereo image de-fencing using smartphones. In: Proceedings of the International Conference

Acoustics, Speech and Signal Processing (ICASSP), pp. 1792–1796 (2017)
11. Xue, T., Rubinstein, M., Liu, C., Freeman, W.T.: A computational approach for obstruction-free photography. ACM Trans. Graph. **34**(4), 1–11 (2015)
12. Liu, W., Mu, Y., Yan, S.: Video de-fencing. IEEE Trans. Circuits Syst. Video Technol. **24**(7), 1111–1121 (2014)
13. Khasare, V.S., Sahay, R.R., Kankanhalli, M.S.: Seeing through the fence: image de-fencing using a video sequence. In: IEEE Proceedings of International Conference on Image Processing (ICIP), pp. 1351–1355 (2013)
14. Negi, C.S., Mandal, K., Sahay, R.R., Kankanhalli, M.S.: Super-resolution de-fencing: simultaneous fence removal and high-resolution image recovery using videos. In: Proceedings of IEEE International Conference on Multimedia and Expo Workshops (ICMEW), pp. 1–6 (2014)
15. Yi, R., Wang, J., Tan, P.: Automatic fence segmentation in videos of dynamic scenes. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 705–713 (2016)
16. Jonna, S., Nakka, K.K., Sahay, R.R.: My camera can see through fences: a deep learning approach for image de-fencing. In: Asian Conference on Pattern Recognition (ACPR), pp. 261–265 (2015)
17. Jonna, S., Nakka, K.K., Sahay, R.R.: Deep learning based fence segmentation and removal from an image using a video sequence. In: Computer Vision—ECCV Workshops, pp. 836–851. Springer, 2016
18. Jonna, S., Voleti, V.S., Sahay, R.R., Kankanhalli, M.S.: A multimodal approach for image de-fencing and depth inpainting. In: International Conference on Advances in Pattern Recognition (ICAPR), pp. 1–6 (2015)
19. Jonna, S., Nakka, K.K., Khasare, V.S., Sahay, R.R., Kankanhalli, M.S.: Detection and removal of fence occlusions in an image using a video of the static/dynamic scene. J. Opt. Soc. Am. **33**(10), 1917–1930 (2016)
20. Yamashita, A., Matsui, A., Kaneko, T.: Fence removal from multi-focus images. In: Proceedings of the International Conference on Pattern Recognition (ICPR), pp. 4532–4535 (2010)
21. Li, Y., Wang, Y., Piao, Y.: Extraction of thin occlusions from digital images. In: Proceedings of SPIE 10255, Selected Papers of the Chinese Society for Optical Engineering Conferences (2017)
22. 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)
23. Mirkamali, S.S., Nagabhushan, P.: Object removal by depth-wise image inpainting. Signal Image Video Process. **9**(8), 1785–1794 (2015)
24. Lee, J., Lee, D.K., Park, R.H.: Robust exemplar-based inpainting algorithm using region segmentation. IEEE Trans. Consum. Electron. **58**(2), 553–561 (2012)
25. Yamashita, A., Tsurumi, F., Kaneko, T., Asama, H.: Automatic removal of foreground occluder from multi-focus images. In: IEEE International Conference on Robotics and Automation, pp. 5410–5416 (2012)
26. Kumar, V., Mukherjee, J., Mandal, S.K.D.: Combinatorial exemplar-based image inpainting. In: International Workshop on Combinatorial Image Analysis, pp. 284–298 (2015)
27. Yamauchi, H., Haber, J., Seidel, H.P.: Image restoration using multiresolution texture synthesis and image inpainting. In: Proceedings of Computer Graphics International, pp. 120–125 (2003)
28. Wei, Y., Liu, S.: Domain-based structure-aware image inpainting. Signal Image Video Process. **10**(5), 911–919 (2016)
29. Farid, M.S., Khan, H., Mahmood, A.: Image inpainting based on pyramids. In: Proceedings of IEEE International Conference on Signal Processing (ICSP), pp. 711–715 (2010)
30. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 886–893 (2005)
31. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. (IJCV) **60**(2), 91–110 (2004)
32. Belongie, S., Malik, J., Puzicha, J.: Matching shapes. In: The 8th ICCV, pp. 454–461 (2001)
33. Freeman, W.T., Roth, M.: Orientation histograms for hand gesture recognition. Technical report, International Workshop on Automatic Face and Gesture Recognition. IEEE Computer Society (1995)
34. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press Inc., Orlando (1983)
35. Dalal, N.: Finding people in images and videos. Ph.D. thesis, Institut National Polytechnique de Grenoble/INRIA (2006)