Journal of the Optical Society of America A
OPTICS, IMAGE SCIENCE, AND VISION

Detection and removal of fence occlusions in an image using a video of the static/dynamic scene

SANKARAGANESH JONNA,[1,*] KRISHNA K. NAKKA,[2] VRUSHALI S. KHASARE,[1] RAJIV R. SAHAY,[1,2] AND MOHAN S. KANKANHALLI[3]

[1]Computational Vision Lab, School of Information Technology, Indian Institute of Technology Kharagpur, West Bengal 721302, India
[2]Department of Electrical Engineering, Indian Institute of Technology Kharagpur, West Bengal 721302, India
[3]School of Computing, National University of Singapore, Singapore 119077, Singapore
*Corresponding author: sankar9.iitkgp@gmail.com

Received 14 July 2016; accepted 26 July 2016; posted 5 August 2016 (Doc. ID 270635); published 2 September 2016

The advent of inexpensive smartphones/tablets/phablets equipped with cameras has resulted in the average person capturing cherished moments as images/videos and sharing them on the internet. However, at several locations, an amateur photographer may be frustrated with the captured images. For example, the object of interest to the photographer might be occluded or fenced. Currently available image de-fencing methods in the literature are limited by non-robust fence detection and can handle only static occluded scenes whose video is captured by constrained camera motion. In this work, we propose an algorithm to obtain a de-fenced image using a few frames from a video of the occluded static or dynamic scene. We also present a new fenced image database captured under challenging scenarios such as clutter, poor lighting, viewpoint distortion, etc. Initially, we propose a supervised learning-based approach to detect fence pixels and validate its performance with qualitative as well as quantitative results. We rely on the idea that freehand panning of the fenced scene is likely to render visible hidden pixels of the reference frame in other frames of the captured video. Our approach necessitates the solution of three problems: (i) detection of spatial locations of fences/occlusions in the frames of the video, (ii) estimation of relative motion between the observations, and (iii) data fusion to fill in occluded pixels in the reference image. We assume the de-fenced image as a Markov random field and obtain its maximum *a posteriori* estimate by solving the corresponding inverse problem. Several experiments on synthetic and real-world data demonstrate the effectiveness of the proposed approach.

OCIS codes: (100.3020) Image reconstruction-restoration; (100.2980) Image enhancement; (100.3190) Inverse problems; (100.2960) Image analysis; (100.0100) Image processing.

http://dx.doi.org/10.1364/JOSAA.33.001917

1. INTRODUCTION

Tourists and amateur photographers capture their cherished moments at historical places or monuments which they visit during their travel. The availability of low-cost smartphones/phablets with sophisticated cameras has led to an increase in the images or videos captured and shared across the internet. Despite the advances in the technology of such devices, sometimes the amateur photographer is frustrated by unwanted elements in the scene. One such hindrance is the presence of barricades or fences occluding the object which the photographer wishes to capture. In recent times, security concerns have resulted in places of tourist interest being barricaded for protection. Fences have become common, restricting access to the public and affecting the aesthetic experience of the tourist who wants to preserve his/her memories for posterity using images/videos. Sometimes fences are essential to protect the spectator from grave danger such as wild animals in zoos. However, one would prefer to enhance the aesthetic appeal of the captured images of these animals by removing interfering fences/barricades.

The problem of image de-fencing [1–4] is basically removal of fences/barricades from an image affected by such occlusions. In this work, we propose a supervised learning-based approach to detect fences/occlusions and an optimization framework to obtain a de-fenced image using a few frames from the video of the occluded scene. The basic idea is to capture a short video clip by moving the camera relative to the occluded scene and use a few frames from it to restore data hidden by the fence in the reference image. A sample frame from a captured video is shown in Fig. 1(a), wherein a fence is occluding parts of a tiger. We also show the third frame from the captured video in Fig. 1(b). As shown with the aid of arrows, we observe that

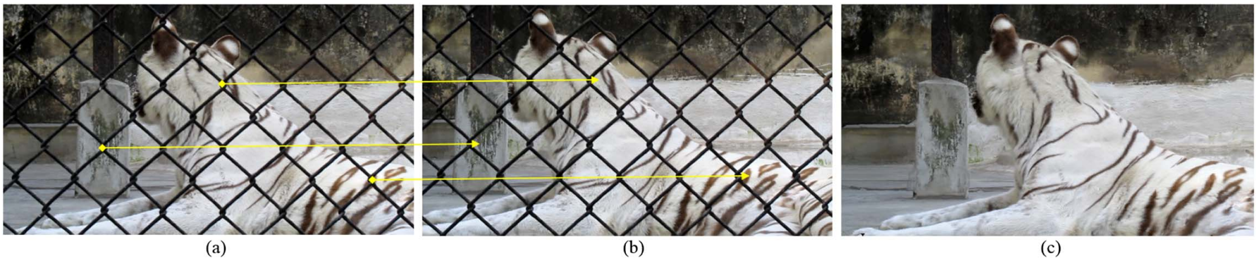


Fig. 1. Image de-fencing: (a), (b) First and third frames from the captured video obtained by moving a camera relative to the occluded scene. (c) De-fenced image corresponding to (a) obtained using the proposed algorithm. See Visualization 1.

relative motion between these frames uncovers data that is hidden behind the fence pixels in Fig. 1(a). This fact can be exploited to perform de-fencing of the degraded reference frame in Fig. 1(a). In Fig. 1(c), we show the output of the proposed algorithm wherein occlusions due to fence pixels have been successfully removed.

We observe that although the problem appears simple, it becomes more challenging when the scene is dynamic and three-dimensional in nature. Our approach for image de-fencing necessitates the solution of three sub-problems, which we identify as (1) automatic detection of spatial locations of fences or occlusions in the frames of the video, (2) estimation of relative motion between the frames, and (3) data fusion to fill in occluded pixels in the reference image with uncovered scene data in additional frames. This is summarized in Fig. 2.

Fence detection is the first task that is addressed in the proposed algorithm. Recently, there has been considerable progress in the area of image/depth inpainting [5–17], whereby most works assume that the spatial location of pixels to be filled in are known *a priori*. It is to be noted that, for the problem at hand, we cannot make such an assumption since the number of fence pixels are too many and it is very tedious to mark them by hand. In our earlier works [3,4], we used the image-matting technique of [18] to extract the foreground fence pixels. However, the drawback of [18] is that it involves *significant* user interaction wherein some pixels of both the foreground (fence) and background are explicitly marked by scribbling. Therefore, here we propose a supervised learning approach to automatically identify fences. In addition, we propose a database of 200 fenced images captured under several challenging scenarios such as clutter, illumination, perspective distortions, etc. We validate the proposed fence-detection algorithm with qualitative and quantitative results.

After the fences/occlusions have been identified, we need to fill in the missing information in order to de-fence the reference frame. A naive idea is to simply inpaint the fenced reference image by a standard image-completion technique. However, such an approach would approximate missing information by propagating neighboring pixel intensities respecting edges/discontinuities in the frame which may fail to accurately reconstruct finely textured regions of the background hidden behind fence occlusions. Importantly, by using inpainting techniques, we do not exploit the important fact that relative motion between the camera and scene can cause additional frames to contain data that is missing in one frame. Similar observations were made by the authors of [1], wherein a video of the fence scene was captured while comparing their work to that of [2], which resorted to image inpainting using the method in [7]. However, in order to exploit the availability of additional data, we need to estimate motion between the frames accurately.

In our previous work [3,4], we assumed that the frames of the input video obtained by panning the occluded scene are shifted *globally*. Hence, we used the affine scale-invariant feature transform (SIFT) [19] image descriptors to match corresponding points in the frames obtained from the captured video. By avoiding false matches with a little user interaction, it is possible to estimate the global pixel motion accurately. The above assumption restricts our algorithms in [3,4] to videos

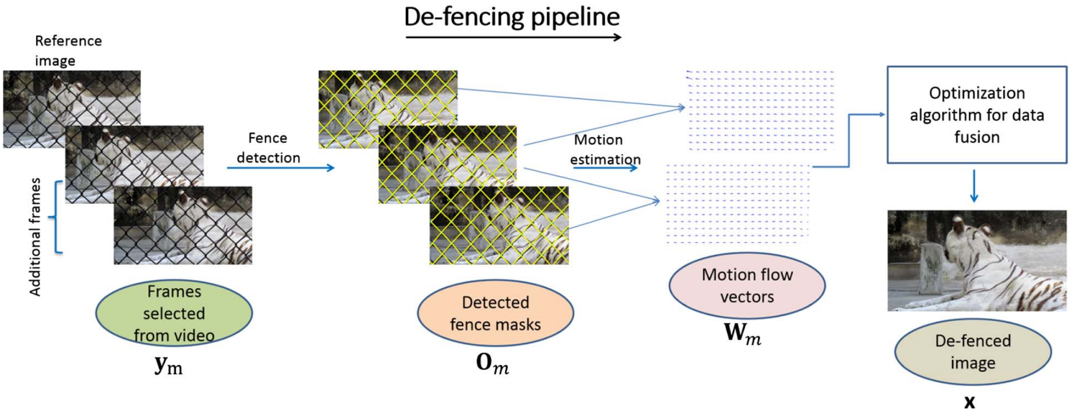


Fig. 2. Schematic for image de-fencing.

containing global motion only, which is not general enough for real-world data wherein scene elements can be dynamic also. Hence, in this work, we used a recently proposed algorithm for estimating optical flow [20].

The final task in our method is fusion of data from additional frames to de-fence the reference image. For this purpose, we use a degradation model to describe the formation of images affected by occlusions due to fences. The de-fenced image is modeled as a Markov random field (MRF). The task of image de-fencing is posed as an inverse problem. We use the loopy belief propagation technique [21] to optimize an appropriately formulated objective function. Our approach is more accurate than mere image inpainting since we use data from neighboring frames to derive the maximum *a posteriori* estimate of the de-fenced image.

A video de-fencing algorithm is proposed in [22]. However, the method of [22] restricts the motion of the camera as affine and users need training to capture the video in an appropriate manner. A major drawback of [22] is that it is limited to defencing only static background scenes. This is because the algorithm of [22] failed to distinguish between the parallax caused by fences as well as dynamic scene elements in the background. We also note that the authors of [22] do not provide a comprehensive evaluation of their algorithm for fence localization/detection.

In a very recent work, an algorithm is proposed in [23] which is capable of removing degradations such as occluding fences, reflections, and raindrops from captured images using a short video sequence. However, the method of [23] imposes unrealistic constraints on the relative motion between the camera and the scene, allowing only approximate horizontal movement without any rotation, similar to capturing a panaroma. Further, the scene is assumed to be roughly static, which is another major drawback. Another limitation of [23] is that the authors do not provide a comprehensive evaluation of their algorithm for fence detection. We address the significant limitations of both [22] and [23] in this work.

Although there have been previous attempts at removing fences from images [1–4,22,23] the novelty of our method is the use of a learning-based approach for fence detection and formulation of an optimization-based framework to fuse data from multiple frames of the input video in order to fill in fence pixels. Importantly, we provide qualitative and quantitative comparison results with an existing fence/lattice detection algorithm [24]. Unlike those in [22,23], the proposed method can handle reasonably arbitrary camera motions such as rotation, zooming, etc., during video capture. Moreover, we can de-fence scenes containing static/dynamic objects. A significant advantage of the proposed approach is that we can de-fence a reference image using only 2 additional frames on average from the captured video in contrast to the 14 frames needed by the method of [22] and the 4 frames used in [23]. We demonstrate the superiority of the proposed algorithm over the state-of-the-art techniques through experiments using both synthetic and real-world data.

This paper is organized as follows. We review methods in the literature for fence detection, inpainting, and image de-fencing in Section 2. In Section 3, we outline details of the proposed methodology, including fence detection and the optimization framework for data fusion. Experimental results and comparisons with the state-of-the-art algorithms are presented in Section 4. Finally, conclusions are given in Section 5.

2. RELATED WORKS

A. Fence Detection

There has been significant research in the field of regular or near-regular pattern detection [1,24,25–27]. Initially, the authors in [25] formulated the computational model for periodic pattern perception based on the theory of frieze and wallpaper groups. Later, the work of [26] posed lattice discovery as a higher-order correspondence problem and discovered patterns with significant texel variations. An effective fence-detection algorithm is proposed in [24]. The authors in [24] detect regular textures in three phases using mean shift belief propagation and regularized thin-plate spline warping. The work in [22] addressed the video de-fencing problem wherein a soft fence-detection method is proposed by using visual parallax as the cue to distinguishing fences from the unoccluded pixels. Recently, the authors in [23] proposed a computational approach wherein they have recovered the occluding foreground along with the restored background using visual parallax.

B. Image Completion/Inpainting

Diffusion-based image inpainting techniques [5,28,29] use smoothness priors to propagate information from known regions to the unknown region. These algorithms work satisfactorily if the region with missing data is small in size and low-textured in nature. On the contrary, exemplar-based techniques [7,9,30,31] fill in the occluded regions using similar patches from other locations in the image. Methods belonging to this category possess the advantage of being able to recreate texture missing in large regions of the image.

Criminisi *et al.* [7] combined the advantages of both structure propagation and texture synthesis for filling in large regions. The work of Patwardhan *et al.* [13] addressed video inpainting under constrained camera motion. The authors of [32] approached the image-inpainting problem using a variational framework employing split Bregman technique. He *et al.* [33] added another novel aspect to exemplar-based inpainting techniques wherein statistics of patch offsets have been exploited. Recently, Ruzic *et al.* [12] proposed a context-aware inpainting algorithm using a normalized histogram of Gabor responses as the contextual descriptors. Ebdelli *et al.* [34] proposed a video inpainting algorithm by considering additional frames to inpaint a reference frame. The method in [35] uses blurred images captured with three different aperture settings to remove thin occluders from images.

C. Image/Video De-fencing

Liu *et al.* in [2] first addressed the image de-fencing problem via inpainting of the occluding foreground pixels. In their method, the fence mask is detected using a regularity-discovery algorithm proposed in [26]. The filling in of the occluded pixels is attempted by using the algorithm of [7]. Basically, [2] treats the de-fencing problem as an inpainting problem. Subsequently, the authors in [1] extended the algorithm in [2] using mutliple images from a captured video, obtaining significant improvement in performance due to availability of hidden information in additional frames. The work in [1] proposed a learning-based algorithm using support vector machine (SVM) to improve the accuracy of lattice detection and segmentation. The work of [22,23] also addressed the problem of video de-fencing but was restricted to handling only roughly static occluded scenes.

In a previous work [3], we proposed an improved multi-frame de-fencing algorithm using loopy belief propagation. However, the work in [3] assumed that motion between the frames was global and used the technique of [18] for fence detection.

3. PROPOSED METHODOLOGY

A. Degradation Model

The image de-fencing problem can be modeled as

$$\mathbf{O}_m\mathbf{y}_m = \mathbf{y}_m^{\text{obs}} = \mathbf{O}_m[\mathbf{W}_m\mathbf{x} + \mathbf{n}_m], \tag{1}$$

where $\mathbf{O}_m$ is the binary fence mask corresponding to the mth frame, $\mathbf{y}_m$ represents the mth frame of the video, $\mathbf{y}_m^{\text{obs}}$ is the mth frame wherein pixels occluded by fences have been excluded using $\mathbf{O}_m$, $\mathbf{W}_m$ is the warp matrix, $\mathbf{x}$ is the de-fenced image, and $\mathbf{n}_m$ is the noise assumed as Gaussian.

As described earlier in Section 1, the problem of image de-fencing was divided into three sub-problems and the overall workflow of the proposed approach is shown in Fig. 2.

B. Fence Detection

The first task is to detect fence pixels corresponding to each observation $\mathbf{y}_m$ in Eq. (1). The fence masks, $\mathbf{O}_m$, in Eq. (1) are used to crop out visible information in the observations. We experimented with several ideas to address this problem and report the results here.

1. Image Segmentation

The simplest approach is to treat fence detection as an image-segmentation problem. We employ the graph-cut segmentation algorithm [36,37,38,39] on the images shown in Figs. 3(a) and 3(f) using the MATLAB wrapper given in [36]. The corresponding segmentation results are shown in Figs. 3(b) and 3(g), respectively. This approach works for simple images with homogeneous backgrounds as in Fig. 3(a). However, for more complex real-world images such as the one shown in Fig. 3(f), the segmentation algorithms in [36] do not work well, as shown in Fig. 3(g).

2. Stroke Width Transform

Interestingly, we observe a similarity between the problem of detecting text in natural images and the task of fence detection. Signboards, advertisement billboards, license plates, etc., generally contain text of roughly constant width and uniform color inside the letter strokes. This property has been exploited for effective text detection in natural images using stroke width transform (SWT) [40]. We employed SWT for fence detection in Figs. 3(a) and 3(f) and observed that the algorithm works to a reasonable extent for scenes with simple homogeneous backgrounds as in Fig. 3(a). However, it fails for many other real-world images with complex backgrounds such as the fenced image in Fig. 3(f). The corresponding result is shown in Fig. 3(h).

3. Gabor Filtering

We also observe that fences commonly seen outdoors, such as in public places, are generally symmetric in shape and exhibit strong edges along certain orientations. To exploit the directional nature of the fences, we employed directional filters existing in the literature such as Gabor filters [41]. In the 2D spatial domain, the Gabor function is a complex exponential modulated by a Gaussian,

$$g_{\theta,\sigma,\lambda,\psi,\gamma}(x,y) = \exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right)\exp\left(i\left(2\pi\frac{x'}{\lambda}+\psi\right)\right), \tag{2}$$

where $x' = x\cos\theta + y\sin\theta$, $y' = -x\sin\theta + y\cos\theta$, x and y denote the pixel positions, θ represents the orientation of the Gabor filter, σ denotes the standard deviation of the Gaussian function, λ is the wavelength, ψ is the phase offset, and γ represents the aspect ratio. Here, θ can be varied between 0° and 360° based on the fence orientation. We applied Gabor filters to the observations shown in Figs. 3(a) and 3(f). We use two Gabor filters with orientation angles $\pi/4$, $-\pi/4$ deg and other parameters fixed as $\lambda = 4$, $\psi = 0$, $\gamma = 0.5$, and $\sigma = 4$.

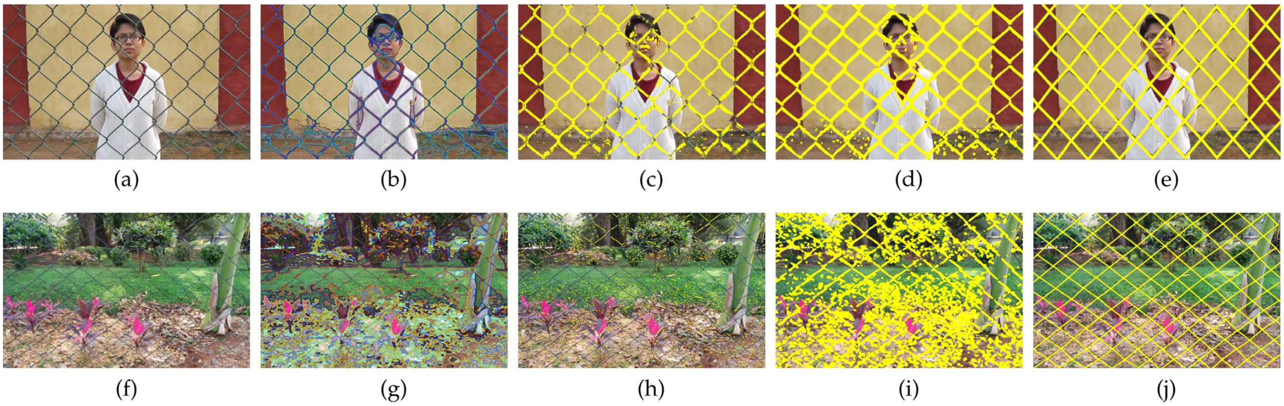


Fig. 3. (a), (f) Two observations chosen from two videos of fenced scenes. (b), (g) Segmentation results obtained using [36] for (a) and (f), respectively. (c), (h) Detected fence pixels using SWT [40]. (d), (i) Estimated fence masks using Gabor filter [41] responses obtained for (a) and (f), respectively. (e), (j) Fence masks obtained for observations (a) and (f) using the proposed algorithm.

As shown in Fig. 3(d), these filters work well only for the case of homogeneous backgrounds. However, as depicted in Fig. 3(i), Gabor filters also respond to texture/strong edges in the background apart from oriented fence texels.

4. Image Matting

If one considers the fence as the foreground and the non-occluded regions in a frame as the background, then we can leverage on the progress made in the area of image matting [18,42] for fence detection. The algorithm in [18] requires a human to mark some foreground and background pixels by means of scribbling, which is used as input to extract an alpha matte. As an illustration, we show the scribbles put by the user on an observation in Fig. 4(a). This scribbled observation is fed as input to the technique in [18], whose output is a gray-scale intensity image representing the alpha matte. We threshold this image to obtain a binary mask that denotes the locations of fence pixels as shown in Fig. 4(b). This approach results in fairly accurate fence masks. However, the major drawback of [18] is that it involves significant user interaction and hence is impractical for use in real-world scenarios.

5. Proposed Supervised-Learning-Based Approach

In many real-world images, fence texels are regular in shape (e.g., rhombic) and have joints at their vertices. To exploit the structural property of such fences, we propose a learning-based approach to detect joint positions in fenced images which are subsequently connected by straight edges. It is amply demonstrated in the literature that histogram of gradients (HoG) [43] features have been successful in many detection and object classification problems. HoG descriptors have several key advantages such as robustness to (a) small changes in image contour locations and directions and (b) significant changes in image illumination and color, remaining as discriminative and separable as possible. Hence, in this work, we extracted HoG features [43] which are used as input to the proposed supervised learning framework. For example, a fenced image taken from a video is shown in Fig. 5(a) and the HoG features corresponding to one fence texel are depicted in Fig. 5(b). Note that we consider a small image patch centered around a joint as the fence texel. The inverse HoG corresponding to Fig. 5(b) obtained using [44] is shown in Fig. 5(c). We observe that HoG features shown in Fig. 5(b) resemble the fence texel shape in the original color image of Fig. 5(a). This characteristic property of HoG features is the reason for its highly discriminative nature, which proves useful for identifying fence texels in occluded images.

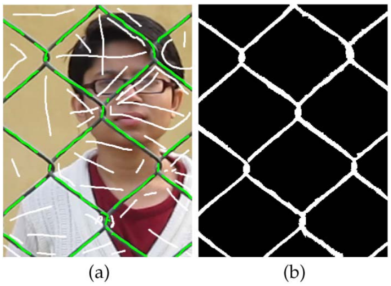


Fig. 4. Fence detection using [18]: (a) Input frame with scribbles marked by a user. (b) Binary fence mask generated by thresholding the gray-scale alpha map obtained using [18].

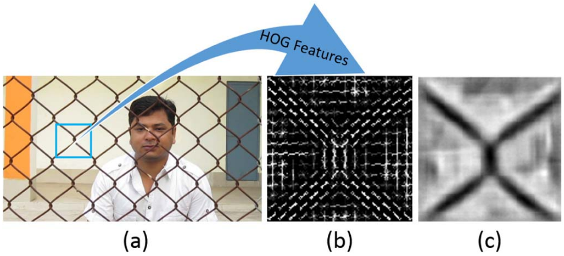


Fig. 5. Visualization of HoG features: (a) Color image with fences. (b) HoG descriptors of one fence texel containing a single joint. (c) Inverse HoG visualization obtained using [44] corresponding to (b).

Supervised learning algorithms critically depend upon features drawn from images. In a recent work [44], it has been cogently argued that output of object-recognition algorithms can be understood and their performance significantly improved by investigating the role of HoG features being extracted from the input images. Therefore, to comprehend why the proposed HoG-based approach should work well in detecting fences in real-world images under challenging scenarios such as clutter, low light, and poor contrast, we show in Fig. 6 positive and negative samples for fences, corresponding HoG features, as well as inverse HoG visualizations. We consider image patches of size 30 × 30 pixels centered around the joints of fences as positive fence texels. Sample fence and non-fence texels are shown in first column of Figs. 6(a) and 6(b). The corresponding HoG features and visualization of the inverse HoG obtained using [44] are shown in the second and third columns, respectively, of Fig. 6. We notice that visualization of inverse HoG reveals the details hidden in the top portion of the fence texel shown in the last row of Fig. 6(a), which is difficult even for humans to discern.

Using a subset of 200 fenced images in the proposed dataset, we manually cropped 30 × 30 pixel-sized sub-images corresponding to fence joints and non-joints to obtain positive

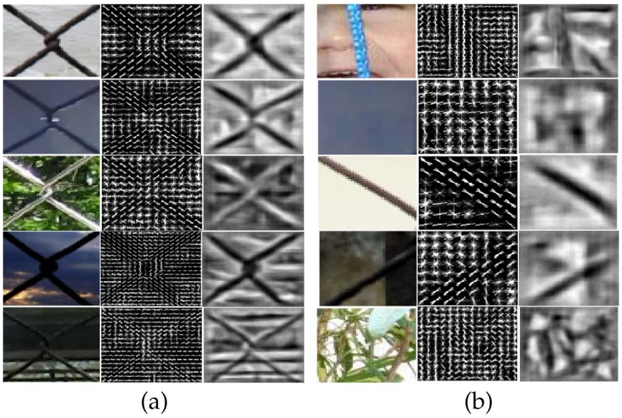


Fig. 6. HoG features and inverse HoG visualization obtained using [44]: (a) First, second, and third columns are sample fence texels, HoG features, and inverse HoG, respectively, obtained using [44]. (b) First, second, and third columns depict sample non-fence texels, HoG features, and inverse HoG, respectively, obtained using [44].

and negative fence texels. The base dataset for training in the proposed supervised-learning algorithm consists of 2000 positive examples of fences as well as 6000 negative samples of non-fence texels. Initially, all the images in the dataset are pre-processed by histogram equalization to reduce the effects of illumination changes. Each training image is divided into non-overlapping cells of size 4 × 4 pixels wherein image gradients are computed in terms of magnitude as well as orientation. At every pixel in the cell, the gradient orientation is quantized into one of the nine bins, weighted based on its magnitude. The orientation bins are evenly spaced over 0°–180°, with each bin of size 20°. For each cell, we compute the histogram of nine orientations to form a feature vector of size 9. A set of four cells is clustered together to form a single block. Each block is thus represented by a feature vector of length 4 × 9 with an overlap of two cells between neighboring ones. Finally, all the $L2$-normalized feature vectors from the blocks are concatenated to obtain a single large feature vector of size 6 × 6 × 4 × 9 corresponding to a single fence/non-fence texel of size 30 × 30 pixels.

Next, we train an SVM with the extracted HoG features to detect whether an input image patch contains a joint of the fence or not. We choose an RBF kernel given as $k(x_i, x_j) = \exp(\gamma \|x_i - x_j\|^2)$, where the parameters γ and misclassification penalty C are found by iterating on a logarithmic grid and optimally selected based on the error rate estimated on a five-fold cross validation [45].

During the test phase, a sliding window is used to densely scan the test image from left to right and top to bottom with a stride of five pixels along both directions. For each detector window of size 30 × 30 pixels in a test image, HoG features are extracted and fed to the trained SVM classifier to detect the presence or absence of a joint of the fence in this sub-image. Now, to connect the detected joints, we calculate inter-joint distances along the horizontal as well as vertical directions. The median of those inter-joint distances is assumed as the dimension of the individual texel. For every detected joint, we find candidates for neighboring joints within a region of half the inter-joint distances upto a reasonable threshold. The false positives from the detected joints are automatically eliminated as those will be located outside the considered search space. Finally, we connect the valid detected joints using straight edges to obtain the fence mask denoted by $\mathbf{O}_m$ in Eq. (1). Note that the proposed approach for fence detection is reasonably robust to perspective distortions and magnification in the fenced images since we are only detecting fence joints rather than the entire rhombic shape/pattern.

C. Motion Estimation

The basic idea behind our method is that occluded image data in the reference frame is uncovered in other frames of the captured video. Relative shifts among the frames have to be estimated in the degradation model of Eq. (1) to effect the image operations corresponding to $\mathbf{W}_m$. In our previous works [3,4], we assumed that the background (non-fence) pixels in the frames are shifted with respect to each other by a globally fixed amount.

However, in real-world videos, the scene can consist of objects which are dynamic. To employ the proposed algorithm for such dynamic videos, we need accurate local pixel motion among the frames. Recently, significant advances have been made in estimating dense optical flow in a robust manner [20,46,47–49]. Brox and Malik [20] proposed an optical flow-estimation technique, wherein descriptor matching is integrated in a variational framework. We use the algorithm of [20] in this work. The objective function formulated in [20] is

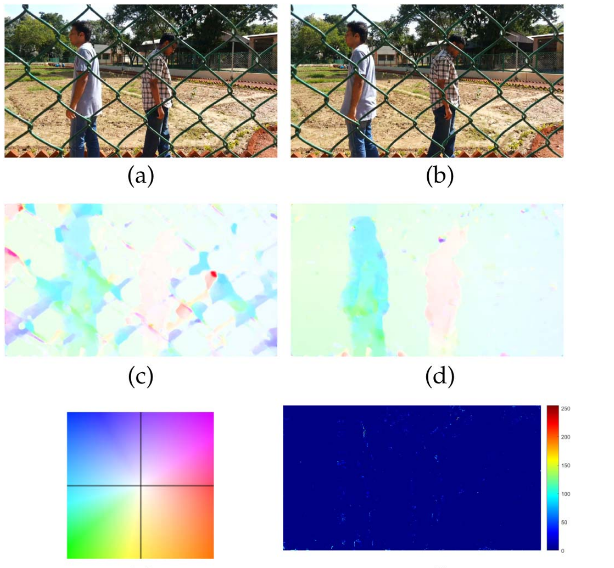
$$E(\mathbf{f}) = E_{\text{color}}(\mathbf{f}) + \gamma E_{\text{gradient}}(\mathbf{f}) + \alpha E_{\text{smooth}}(\mathbf{f}) + \beta E_{\text{match}}(\mathbf{f}, \mathbf{f1}) + E_{\text{desc}}(\mathbf{f1}), \tag{3}$$

where the first term E_{color} encodes the common assumption that corresponding points should have same color. The second term E_{gradient} enforces the gradient constraint and the third term E_{smooth} adds the regularization constraint; E_{match} and E_{desc} integrate the point correspondences from descriptor matching and matched descriptors, respectively. The symbols α, β, and γ are tuning parameters which can be determined empirically. Optical flow at points is denoted by $\mathbf{f}{:=}(u, v)^T$, and $\mathbf{f1}$ represents correspondence vectors obtained by descriptor matching at some pixels. In [20], Eq. (3) is expanded as follows:

$$\begin{aligned} E(\mathbf{f}) = {} & \int_{\Omega} \psi(|\mathbf{y}_2(\Theta + \mathbf{f}(\Theta)) - \mathbf{y}_1(\Theta)|^2) \mathrm{d}\Theta \\ & + \gamma \int_{\Omega} \psi(|\nabla \mathbf{y}_2(\Theta + \mathbf{f}(\Theta)) - \nabla \mathbf{y}_1(\Theta)|^2) \mathrm{d}\Theta \\ & + \alpha \int_{\Omega} \psi(|\nabla u(\Theta)|^2 + |\nabla v(\Theta)|^2) \mathrm{d}\Theta \\ & + \beta \int \delta(\Theta) \rho(\Theta) \psi(|\mathbf{f}(\Theta) - \mathbf{f1}(\Theta)|^2) \mathrm{d}\Theta \\ & + \int \delta(\Theta)(|\mathbf{s}_2(\Theta + \mathbf{f}_1(\Theta)) - \mathbf{s}_1(\Theta)|^2) \mathrm{d}\Theta, \end{aligned} \tag{4}$$

where $\mathbf{y}_1$, $\mathbf{y}_2{:}(\Omega \in \mathbb{R}^2) \rightarrow \mathbb{R}^d$ are two observations to be aligned, d is number of channels, and the function $\psi(s^2) = \sqrt{s^2 + \varepsilon^2}$, $\varepsilon = 0.001$ is used to deal with occlusions and other deviations of the matching criterion. Variable $\Theta{:=}(x, y)^T$ denotes a point in the image domain $\mathbf{\Omega}$ and $\mathbf{s}_1$, $\mathbf{s}_2$ denote the sparse fields of descriptor feature vectors in frames 1 and 2, respectively.

The delta function $\delta(\Theta)$ indicates if a descriptor match is available in location Θ and $\rho(\Theta)$ denotes confidence of the match. Descriptor matches are obtained by matching densely sampled HoG features in the two images $\mathbf{y}_1$, $\mathbf{y}_2$. In this work, we estimate the optical flow by minimizing the energy function in Eq. (4) using the algorithm proposed in [20]. When the optical flow for frames extracted from a video of a fenced scene are estimated by [20], we observe erroneous values around the fenced or occluded pixels. To avoid these errors, we blur only the fences in the observations prior to computing optical flow using [20]. Depending on the thickness of the fences in the input frames, we use a Gaussian kernel with standard deviation (σ) varying from 1 to 2. In Figs. 7(a) and 7(b), we show two frames from a video depicting persons walking in opposite directions. In Figs. 7(c) and 7(d), respectively, we show the estimated optical flow before and after blurring the fences in the observations given in Figs. 7(a) and 7(b). We observe a significant improvement in the estimated optical flow [Fig. 7(d)] after



(a) (b) (c) (d) (e) (f)

Fig. 7. (a), (b) Frames from a real-world video. (c), (d) Optical flow between (a) and (b) estimated without and with blurring of fences in (a) and (b) before using [20]. (e) Optical flow color coding [50]. (f) Residual error map between (a) the frames and (b) backwarped frame.

the fences have been blurred using a Gaussian kernel, $\sigma = 2$, prior to using [20]. In Fig. 7(e), we show the color-coding scheme employed in Figs. 7(c) and 7(d), wherein flow vector direction is coded by hue and length by saturation. To depict the accuracy of the estimated optical flow shown in Fig. 7(d), we use it to obtain the residual error map between the reference frame in Fig. 7(a) and the backwarped frame of Fig. 7(b) as Fig. 7(f). The error map in Fig. 7(f) has a root mean square error (RMSE) of 0.018 with only minor artifacts at the contours of bodies of the two individuals in the scene.

D. Optimization Using Loopy Belief Propagation

Once relative pixel shifts between the frames of the captured video are estimated as just described, we need to fuse image data in order to fill in pixels in the reference image that are occluded by fences. In computer vision, contextual constraints are widely used to solve inverse problems. Such constraints are modeled by well-known graphical models, e.g., Markov random fields. In this work, we also propose to model the de-fenced image as a Markov random field. We formulate an optimization framework for obtaining its maximum *a posteriori* estimate as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \sum_{m=1}^{K} \|\mathbf{y}_m^{\text{obs}} - \mathbf{O}_m \mathbf{W}_m \mathbf{x}\|^2 + \lambda \sum_{c \in C} \mathbf{V}_c(\mathbf{x}), \quad (5)$$

where λ is the regularization parameter, K denotes the number of frames used, c is a clique, and C is the set of all possible cliques. The joint pdf of the MRF can be specified as Gibbsian by the Hammersley–Clifford theorem [51],

$$P(\mathbf{x}) = \frac{1}{Z} \exp(-\mathbf{V}_c(\mathbf{x})), \quad (6)$$

where Z is the partition function and $\mathbf{V}_c(\mathbf{x})$ is the clique potential function. We choose a robust form for the clique potential function as $\mathbf{V}_c(\mathbf{x}) = |x_{i,j} - x_{i,j+1}| + |x_{i,j} - x_{i,j-1}| + |x_{i,j} - x_{i-1,j}| + |x_{i,j} - x_{i+1,j}|$, considering a first-order MRF neighborhood.

Loopy belief propagation (LBP) [21] is a popular message-passing algorithm for inference problems on graphical models. Let P be the set of pixels in the de-fenced image $\mathbf{x}$ and L be a finite set of labels. The labels correspond to quantities that we want to estimate at each pixel, such as intensities. A labeling f assigns a label $f_p \in L$ to each pixel $p \in P$, where $L = \{0, \ldots, 255\}$ and $P = \{0, MN - 1\}$ for an $M \times N$ pixel-sized grid. We assume that the labels should vary slowly almost everywhere but may change dramatically at discontinuities. The quality of a labeling is given by an energy function,

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in N} W(f_p, f_q), \quad (7)$$

where N represents the first-order neighborhood. The first term in Eq. (7) is the data cost defined as $D_p(f_p) = (f_p - \mathbf{y}_m^{\text{obs}}(p))^2$ if p is a visible or non-fence pixel; otherwise, $D_p(f_p) = 0$. The term $W(f_p, f_q)$ measures the cost of assigning labels f_p and f_q to two neighboring pixels, and is normally referred to as the discontinuity cost. In this work, we assume $W(f_p, f_q) = \mathbf{V}_c(\mathbf{x}) = |x_{i,j} - x_{i,j+1}| + |x_{i,j} - x_{i,j-1}| + |x_{i,j} - x_{i-1,j}| + |x_{i,j} - x_{i+1,j}|$ for a first-order MRF neighborhood.

Loopy belief propagation is based on the principle of message passing. Each message is a vector of dimension L. The first step in LBP is message updating, which is followed by belief computation. Message updating is done repeatedly either for T iterations or until each node agrees with the opinion from its neighboring nodes:

$$m_{pq}^t(f_q) = \min_{f_p}(D_p(f_p) + W(f_p, f_q) + \sum_{s \in N(p)-q} m_{sp}^{t-1}(f_p)), \quad (8)$$

where $N(p) - q$ denotes the neighbors of p other than q. After T iterations, a belief vector is computed for each node,

$$b_q(f_q) = D_q(f_q) + \sum_{p \in N(q)} m_{pq}^T(f_q). \quad (9)$$

Finally, the label f_q^* that minimizes $b_q(f_q)$ individually at each node is selected.

E. Algorithm

Algorithm 1. Image de-fencing algorithm

```
Input: $\mathbf{O}_m$, $\mathbf{y}_m^{\text{obs}}$, $\lambda$, $L$
Initialize all messages $m_{pq}(f_q)$ to zero
while $t \le T$ do
    for $f_q = 1: L$ do
        $m_{pq}^t(f_q) = \min_{f_p}(D_p(f_p) + W(f_p, f_q)$
            $+ \sum_{s \in N(p)-q} m_{sp}^{t-1}(f_p))$
        $t \leftarrow t + 1$
for $f_q = 1: L$ do
    $b_q(f_q) = D_q(f_q) + \sum_{p \in N(q)} m_{pq}^T(f_q)$
for $q = 0: MN - 1$ do
    $\mathbf{x}(q) = \arg\min_{f_q} b_q(f_q)$
```

4. EXPERIMENTAL RESULTS

In this section, initially we report both qualitative and quantitative results of our proposed fence-detection algorithm on various datasets. Subsequently, we report the de-fencing results obtained using our optimization framework on synthetic and real-world videos. To validate our proposed approach, we compare with the state-of-the-art inpainting as well as de-fencing techniques. We used only three frames from each captured video for all the image de-fencing results reported here using the proposed algorithm. The de-fencing procedure is carried out individually in each color channel and the results combined to generate the de-fenced RGB color image. For all our experiments, we fixed λ value as 0.0005 in Eq. (4). We ran all our experiments on a 3.4 GHz Intel Core i7 processor with 8 GB of RAM. The execution time of our non-optimized MATLAB implementation is of the order of a few tens of seconds.

A. Validation of Proposed Algorithm for Fence Detection

For evaluating our supervised learning-based algorithm we used two different datasets. First, we collected a dataset consisting of 200 real-world images under diverse scenarios and complex backgrounds using a mobile phone camera. These real-world images and videos were captured under several challenging conditions such as poor illumination, clutter, perspective distortion due to non-frontal camera viewpoint, and unconstrained free-hand movement.

Secondly, we used a subset of images from the Penn State University near-regular texture (PSU NRT) database [52]. The images in the NRT database are divided into three categories. Dataset 1 ($D1$) contains 67 images with opaque texels and appearance variations of the repeating elements due to different viewpoint and lighting conditions. Dataset 2 ($D2$) of the NRT database contains 73 images with see-through or wiry structures. Dataset 3 ($D3$) consists of 121 images of city buildings containing multiple repeating patterns with perspective distortion. However, only a subset of 40 images from $D2$ are of fences. We report qualitative and quantitative results of the proposed machine-learning approach for detection of fences in these 40 images of the $D2$ dataset in the NRT database. The images in the NRT database are provided with corresponding ground truth. For our proposed database of fenced images, we used the matting technique in [18] to generate the ground truth fences.

As discussed in Section 3, an SVM classifier was trained using HoG features extracted from a dataset of 8000 (2000 positive fence texels and 6000 non-fence texels) sub-images of resolution 30 × 30 pixels obtained only from a subset of images in the proposed fenced image database. We compare the results of [24] with the output of our algorithm on both the PSU NRT dataset [52] and the proposed fenced-image dataset. Initially, in Fig. 8, we show qualitative comparison results for some images from the PSU NRT dataset [52]. For the images shown in the first and third columns of the first row of Fig. 8, we show that the method of [24] is only partially successful in estimating fence pixels, whereas in the second and the fourth columns, we observe that the proposed method is able to extract the fences completely. However, for the images shown in the first and third columns of the second and third rows of Fig. 8, we see that the method of [24] completely failed to detect fence pixels, whereas in the second and the fourth columns, we observe that the proposed method is able to extract the fences accurately over most of the image regions. For the image shown in second row, second column of Fig. 8 we note that, due to the fence being occluded by hands and hair, the proposed algorithm could not properly detect fence pixels at the bottom of the image. We also acknowledge that there are a few minor misdetections of fence pixels in the images shown in the second row, fourth column and third row, second column of Fig. 8. Interestingly, we observe that our method performs successfully even when the fences are deformed in shape due to non-fronto-parallel viewpoint of the camera as shown in the image in the first row and fourth column of Fig. 8. This is because the joint positions are not highly sensitive to perspective distortions and they remain stable in shape even when the camera viewpoint is changed.

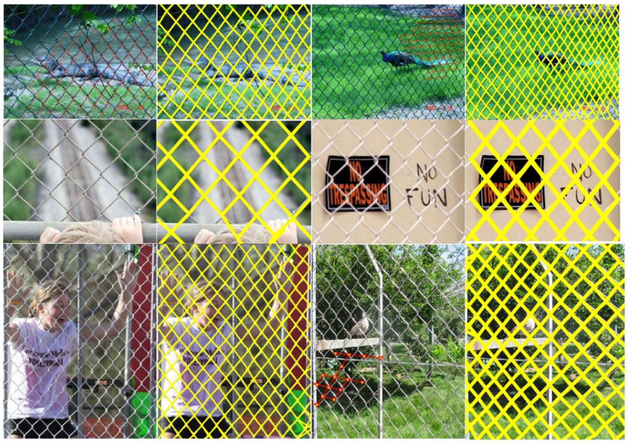


Fig. 8. Sample fence detection results on PSU NRT data set [52] (this figure is best seen in color in the electronic version). Row 1 shows sample results where algorithm in [24] partially detected the fence pixels while the proposed method succeeds in detecting the entire fence. Rows 2 and 3 show sample results where the method of Park *et al.* [24] fails totally while ours succeeds. For each pair of images shown, the left image is the result of [24] and the right image is the result of the proposed HoG + SVM method.

We found that on 70 images of the proposed fenced image dataset, the algorithm in [24] detected fences only partially. We show some sample results to illustrate this fact in the first row of Fig. 9. For the rest of the 130 images in our dataset, the method of [24] completely failed to detect any fence texels. Some of these challenging images are shown in rows 2–5 of Fig. 9. For each pair of images shown, the left picture is the result of Park *et al.* [24] and the right image shows the result of the proposed algorithm. We acknowledge that, due to poor illumination and low contrast, our algorithm failed to detect fences at some portions of the images given in the second row, fourth column; third row, fourth column; and fifth row, second column of Fig. 9.

For the sake of comparison, we also trained an SVM classifier by extracting Gabor features. Gabor features are extracted by convolving each image in the training data set (which was identical to the database from which we extracted HoG features) with 40 Gabor kernels. Subsequently, we trained

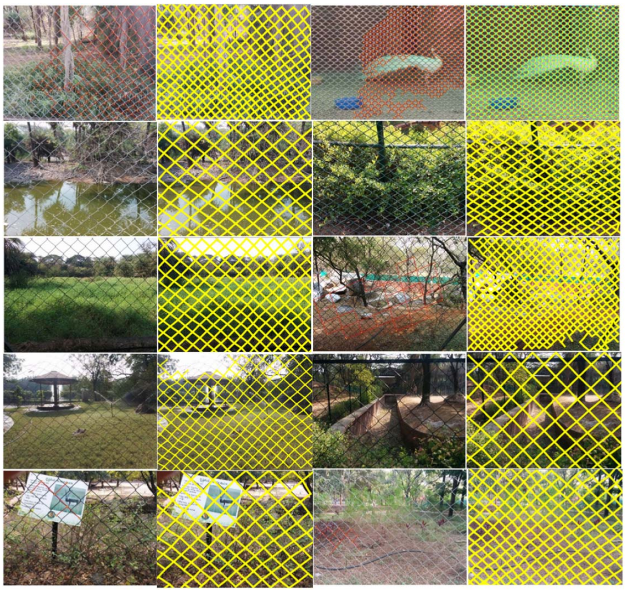


Fig. 9. Sample fence detection results on the proposed fenced image data set (this figure is best seen in color in the electronic version). Row 1 shows sample results where the algorithm in [24] partially detected the fence pixels while proposed method succeeds in detecting the complete fence. Rows 2–5 show sample results where the method of [24] fails completely while ours succeeds. For each pair of images shown, the left image is the result of [24] and the right image is the result of the proposed HoG + SVM method.

an SVM classifier which gave moderate performance compared to [24] and the proposed HoG + SVM technique as reported in Table 1.

We quantitatively compare the proposed approach by calculating different performance metrics such as precision, recall, and F-measure on the two datasets. Precision is the percentage of detected texels that are correct,

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{10}$$

and recall is the percentage of correct texels that are detected,

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{11}$$

where TP, FP, and FN denote true positive, false positive, and false negative values, respectively. And finally, a combined measure that assesses the precision–recall tradeoff is the F-measure, which is the weighted harmonic mean specified as

$$F\text{-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{12}$$

The performance of various techniques on the PSU NRT [52] dataset and the proposed fenced-image dataset has been reported in Table 1. For the PSU NRT dataset [52], the F-measures obtained for [24] and the Gabor + SVM approaches are 0.62 and 0.76, respectively. On this dataset, the F-measure for the proposed method (HoG + SVM) is 0.93. For the proposed fenced-image dataset, the F-measures obtained for [24] and the Gabor + SVM approaches are 0.41 and 0.79, respectively. On this dataset, the F-measure for the proposed method (HoG + SVM) is 0.96.

B. Image De-fencing

Initially, we reported results of the proposed method for image de-fencing on synthetic data. Here, we assumed that the locations of fence pixels were known and that the fence was static. Only the background is shifted with respect to the static fence in the different observations. We assumed that the shift of the background pixels was global and known *a priori*. We generated synthetically four frames by shifting the original image with displacements of (−8, −8), (6, 6), and (15, 15) pixels. This is evident in Fig. 10(a), wherein the optical flow vectors between synthetically generated first and fourth observations are overlaid on the first frame. The thickness of the fence is chosen as 15 pixels. The de-fenced images obtained using state-of-the-art image inpainting algorithms [7] and [32] are shown in Figs. 10(b) and 10(c), respectively. Herein, we observe that the fence patterns are still present in the inpainted images. To analyze the overall performance of the proposed approach, we evaluate the quality of the obtained de-fenced results by varying the number of frames used and thereby relative shifts between the frames. Initially, we experimented with one frame only and the obtained de-fenced result with peak signal to noise ratio (PSNR) of 12.84 dB is shown in Fig. 10(d). In Fig. 10(e), we show the de-fenced image having PSNR of 18.34 dB using two input frames with relative shift of eight pixels between them. Subsequently, we used three frames with the relative shifts between them being chosen as 14 pixels. The corresponding de-fenced image with PSNR of 34.32 dB is shown in Fig. 10(f). Finally, we ran the proposed method with four frames which are relatively shifted by 24 pixels. We show the obtained de-fenced image, which has a PSNR of 38 dB, in Fig. 10(g). Note that there are hardly any artifacts and the fence has been successfully filled in. We can conclude that the quality of the defenced image in terms of PSNR significantly improves with increase in the number of input frames. The corresponding plot is depicted in Fig. 10(i).

Table 1. Quantitative Evaluation of Fence Detection

	PSU NRT Database [52]			Our Database		
Method	**Precision**	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
Park *et al.* [24]	0.95	0.46	0.62	0.94	0.26	0.41
Gabor filter + SVM	0.86	0.68	0.76	0.78	0.80	0.79
Proposed method (HoG + SVM)	0.95	0.92	**0.93**	0.96	0.95	**0.96**

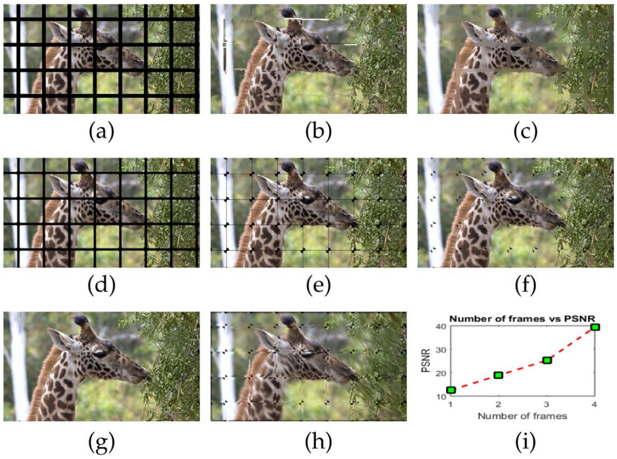


Fig. 10. Image de-fencing (synthetic case): (a) First frame (image courtesy of and copyright Zoological Society of San Diego; used with permission) [53] used in our method. (b), (c) De-fenced images obtained using [7] and [32], respectively. (d)–(g) De-fenced images obtained by the proposed algorithm by using one, two, three, and four observations, respectively. (h) De-fenced image obtained using proposed algorithm with wrong estimates of motion between the frames. (i) Plot showing gain in PSNR with increase in the number of frames.

The quantitative metrices such as RMSE, PSNR, and structural similarity index (SSIM) are given in Table 2. We observe that the PSNR value of the de-fenced image estimated using the proposed algorithm with four input frames is almost twice the PSNR of the inpainted images [Figs. 10(b) and 10(c)]. The de-fenced image obtained using our algorithm is almost identical to the original image. In Fig. 10(h), we show the de-fenced image obtained using the proposed technique when the relative motion between the observations is wrongly given as (−4, −4), (4, 4), and (8, 8). Observe that undesired artifacts appear and the fence is not completely removed in Fig. 10(h).

Table 2. Quantitative Evaluation of Inpainting and De-fencing

Algorithm	PSNR	RMSE	SSIM
Exemplar-based inpainting [7]	19.89	4.76	0.85
Total variation inpainting [8]	23.45	4.63	0.90
Our method (using 4 frames)	**39.37**	**1.22**	**0.99**

Next, we conducted an experiment with real-world data, wherein we have used a video of a song from a movie available on YouTube. In Fig. 11(a), we show the first frame from the captured video overlaid with optical flow vectors with respect to the third observation estimated using [20]. Note that the estimated optical flow depicts complex unconstrained motion of the head of the person relative to the camera. Initially, we used the proposed supervised-learning algorithm to detect the fence pixels in each of the three frames chosen from the video. There is a possibility of missing fence joints/pixels near the boundaries of the image if we do not account for this problem. To deal with fence pixels near the boundaries, we have padded the input image and predict the location of missing joints by considering the dimensions of the fence texels nearest to the boundary. In Fig. 11(b), we show the padded observation in Fig. 11(a) along with fence joints detected using the proposed algorithm. Consider the rhombic shape inside the red-colored box in Fig. 11(b). We have already detected 3 corners of this rhombus in Fig. 11(b) but the fourth vertex lies outside the boundary of the image. We locate the coordinates of the fourth corner in the padded region by considering the dimensions of the partial rhombus inside the red-colored bounding box. The predicted fourth vertex is shown with a cyan-colored cross in Fig. 11(c). We repeat this procedure for all fence pixels lying near the boundaries of the input image. Finally, we join the detected fence joints lying inside the image and the predicted joints lying outside the boundaries in the padded region with straight edges to obtain the fence mask shown in Fig. 11(d).

Finally, the de-fenced image corresponding to the reference frame of Fig. 11(a) obtained using the proposed algorithm is shown in Fig. 11(e). Note that the de-fenced image is reconstructed accurately over the face despite large and complex motion among the frames. Importantly, observe that unlike the methods in [22,23], the proposed de-fencing algorithm can handle this video wherein only the individual in the background was moving rapidly. Since there is no parallax between the fence and the background wall (except the face), there is little contribution of the data fidelity term in Eq. (5) and the fence occlusions are filled in only by the action of the smoothness term. This causes minor blurring/smudging artifacts on the neck, clothing of the person, and also, in some places, in the background region.

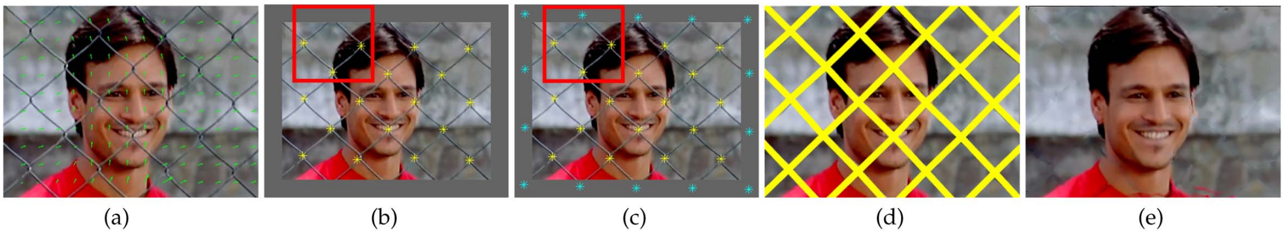


Fig. 11. (a) One frame chosen from a video (image courtesy of and copyright Yash Raj Films Pvt. Ltd.; used with permission) [54]. (b) Detected joints of fence texels using our learning-based algorithm before handling the boundary issue. (c) Detected and predicted fence joints after handling the boundary issue. (d) Obtained fence mask by connecting the detected joints in (c). (e) De-fenced image corresponding to (a) obtained using the proposed algorithm. See Visualization 2.

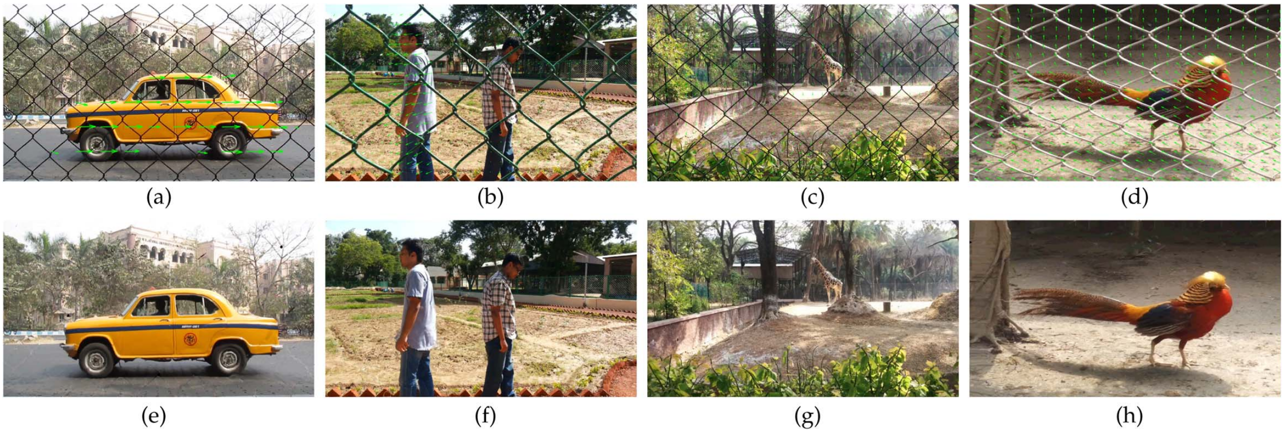


Fig. 12. First row: Images taken from videos. Second row: Corresponding de-fenced results using the proposed algorithm. See Visualization 3, Visualization 4, Visualization 5, and Visualization 6.

Next, we conducted more experiments with several real-world videos containing dynamic background objects. One of the input frames from four different video sequences is shown in Figs. 12(a)–12(d). Optical flow vectors relative to one of the other observations in the video are superimposed on each of the respective frames. Due to the non-global motion among the background objects, we used the method of [20] to estimate the pixel motion among the four observations. The fence pixels in each of these observations shown in Figs. 12(a)–12(d) are detected using the proposed learning-based approach. The corresponding de-fenced images obtained using the proposed algorithm are shown in Figs. 12(e)–12(h), respectively. We observe that, unlike those in [22,23], the proposed algorithm has effectively reconstructed data even for dynamic real-world video sequences. Also note that, for all the results shown in Figs. 12(a)–12(d), we used only three observations from the captured video. We acknowledge that due to insufficient relative motion between the input frames, in Fig. 12(e) we observe minor blurring artifacts/residual fences over certain parts of the reconstructed image.

To demonstrate the robustness of the proposed algorithm across different kinds of camera motion such as translation, rotation, zooming, etc., we have conducted one more experiment using real-world data. In Figs. 13(a) and 13(b), we show two frames chosen from a video captured by moving the camera arbitrarily across the scene. The frame shown in Fig. 13(b) is zoomed and rotated with respect to the reference frame of Fig. 13(a). Note that in this video the fence texels are rectangular in shape. Hence, to detect fence pixels, we trained a different SVM model with a training dataset of images containing rectangular fence texels. The de-fenced image obtained using the proposed algorithm is shown in Fig. 13(c).

C. Comparisons with the State of the Art

To demonstrate the efficacy of the proposed method, we provide comparisons with the state-of-the-art inpainting [32] as well as image de-fencing techniques [22,23]. A frame from a video is shown in Fig. 14(a). The inpainted result is shown in Fig. 14(b) using the method of [32], wherein we observe that residues of fences are still present. In contrast, the fence has been removed completely and there are no residues in the result shown in Fig. 14(c), which was obtained using our technique. Next, we compare our algorithm with the recently proposed video de-fencing method in [22], which used

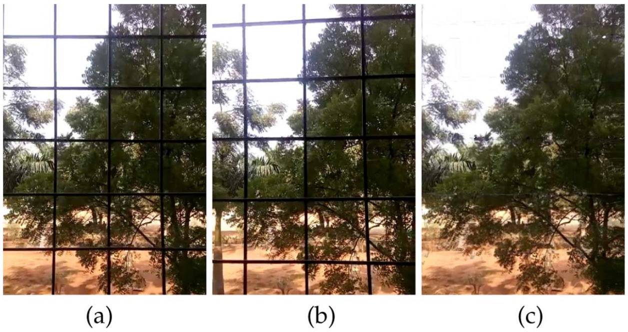


Fig. 13. Video captured with free-hand motion of the camera: (a), (b) Observations obtained from a video exhibiting zoom and rotation. (c) De-fenced image obtained using the proposed algorithm. See Visualization 7.

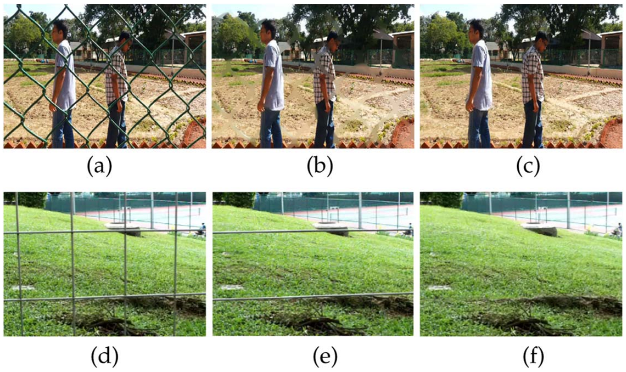


Fig. 14. (a), (d) Frames taken from video sequences. (b) Inpainted result corresponding to (a) obtained using [32]. (c) De-fenced image corresponding to (a) using the proposed algorithm. (e) De-fenced result obtained using [22]. (f) De-fenced result corresponding to (d) using the proposed algorithm.

visual parallax as a cue for fence detection. In Fig. 14(d), we show a frame from a video used by [22]. The approach of [22] failed to detect the horizontal fence structures due to absence of any relative motion in the vertical direction between the neighboring frames of the video. Fig. 14(e) shows the restored frame obtained by the method in [22], where the horizontal bars are still present in the de-fenced image. The de-fenced result obtained using the proposed method is shown in Fig. 14(f), wherein all occluding fences are removed. To compare with the very recent approach of Xue *et al.* [23], we used video sequences named "fence1" and "fence4" from their work, which are shown in Figs. 15(a) and 15(d). The de-fenced images obtained using the method of [23] are shown in Figs. 15(b) and 15(e); in Figs. 15(c) and 15(f), we show the de-fenced images obtained using the proposed algorithm. For the video sequence "fence1," both methods produced the comparable results shown in Figs. 15(b) and 15(c). However, for the case of the video sequence "fence 4," the de-fenced image obtained using the method in [23] is distorted at some places, which is apparent in the close-up of the image in the inset of Fig. 15(e). In contrast, the fence has been removed completely with hardly any distortions in the result shown in Fig. 15(f), which was obtained using our algorithm. Since we use only three frames from the videos, our method is more computationally efficient than [22,23], which uses 5 and 15 frames, respectively.

It is possible to detect fences at various resolutions in the observations using our supervised learning-based algorithm. To illustrate this, we crop a portion of size 250 × 400 pixels from the original frame shown in Fig. 15(d), wherein both the fence closer to the camera and the one at greater depth in the scene are present. We scale this cropped region by zoom factors of $\zeta = [0.2 : 0.2 : 2.2]$, i.e., we explored all magnification factors from 0.2 to 2.2 in steps of 0.2. We report detections of fence joints using the proposed supervised-learning technique on the scaled images at a few magnification factors in Figs. 16(a)–16(c), respectively. Note that we are able to detect joints in the fence closer to the camera and also most of the joints in the fence at the farther depth. We merged all detected fence joints at each resolution and show the final estimated masks for both fences at original resolution ($\zeta = 1$) of the cropped image in Fig. 16(d).

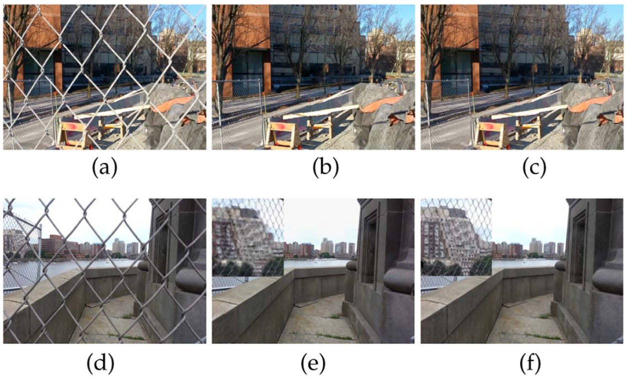


Fig. 15. Comparisons using the video sequences reported in [23]. (a), (d) Frames taken from video sequences. (b), (e) De-fenced results corresponding to (a) and (d) obtained using [23]. (c) and (f) De-fenced image obtained using the proposed algorithm.

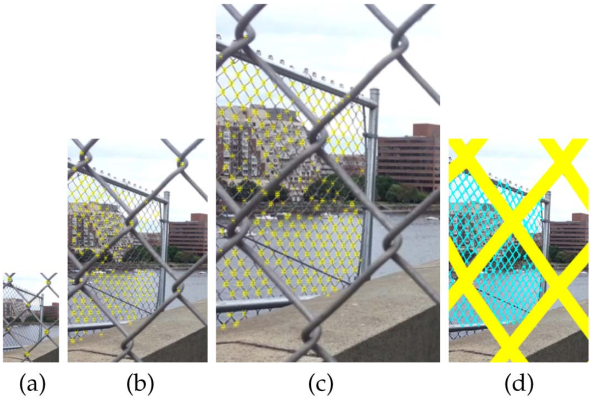


Fig. 16. (a)–(c) Detection of fence joints on images scaled globally by factors $\zeta = 0.4$, 1, and 1.6, respectively. (d) Estimated masks for both fences at original resolution ($\zeta = 1$) of the cropped image.

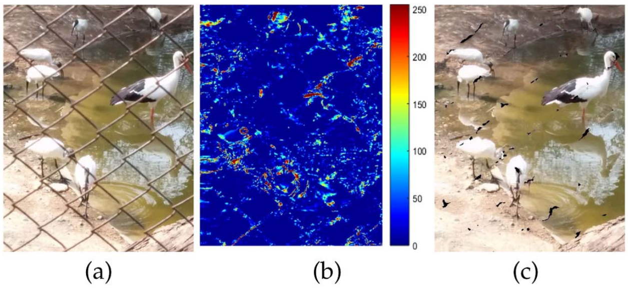


Fig. 17. Failure case: (a) A frame taken from a video. (b) Residual error map between first and backwarped second frame. (c) De-fenced image using our algorithm. See Visualization 8.

D. Failure Case

The proposed de-fencing algorithm fails to de-fence an occluded scene if there are significant errors in motion estimation between the frames. In Fig. 17(a), we show an image of moving birds occluded behind a fence taken from a YouTube video. We chose a few frames from the video and estimated the optical flow using [20] between them. The error between the reference image and the second frame backwarped with estimates of relative motion using [20] is shown in Fig. 17(b). Several significant errors in estimation of relative motion can be observed in Fig. 17(b). The de-fenced image corresponding to Fig. 17(a) but obtained using the proposed approach is shown in Fig. 17(c). We observe that the de-fenced image contains undesired artifacts due to wrongly estimated local motion of birds.

5. CONCLUSION

We propose an approach for de-fencing an image using multiple frames from a video captured by a camera undergoing arbitrary relative motion with respect to a static/dynamic scene. Our approach for image de-fencing necessitates the solution of three sub-problems, which we identified as (a) automatic detection of spatial locations of fences in the frames of the video, (b) accurate estimation of relative motion between the frames, (c) data fusion to fill in occluded pixels in the reference image with uncovered scene data in additional frames. We have proposed a machine-learning approach for the first sub-problem. To validate the accuracy of our proposed learning-based algorithm for fence detection, we have proposed a challenging fenced-image dataset containing 200 real-world images. An optimization-based approach was formulated for fusing data from multiple relatively shifted frames to fill in missing data due to fence occlusions. Our results for both synthetic and real-world data show the effectiveness of the proposed algorithm. We also compared our algorithm with state-of-the-art image de-fencing techniques as well as image-inpainting methods. We believe that a real-time automatic image de-fencing algorithm will be useful, especially with the advent of "smart" computational cameras.

REFERENCES

1. M. Park, K. Brocklehurst, R. T. Collins, and Y. Liu, "Image de-fencing revisited," in *Asian Conference on Computer Vision* (2010), pp. 422–434.
2. Y. Liu, T. Belkina, J. Hays, and R. Lublinerman, "Image de-fencing," in *IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2008), pp. 1–8.
3. V. Khasare, R. Sahay, and M. Kankanhalli, "Seeing through the fence: image de-fencing using a video sequence," in *IEEE International Conference on Image Processing* (IEEE, 2013), pp. 1351–1355.
4. C. S. Negi, K. Mandal, R. R. Sahay, and M. S. Kankanhalli, "Super-resolution de-fencing: simultaneous fence removal and high-resolution image recovery using videos," in *IEEE International Conference on Multimedia and Expo Workshops* (IEEE, 2014).
5. M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *ACM SIGGRAPH* (ACM2000), pp. 417–424.
6. A. Bugeau, M. Bertalmio, V. Caselles, and G. Sapiro, "A comprehensive framework for image inpainting," IEEE Trans. Image Process. **19**, 2634–2645 (2010).
7. A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," IEEE Trans. Image Process. **13**, 1200–1212 (2004).
8. P. Getreuer, "Total variation inpainting using split Bregman," Image Process. On Line **2**, 147–157 (2012).
9. J. Hays and A. A. Efros, "Scene completion using millions of photographs," ACM Trans. Graph. **26**, 1–7 (2007).
10. M. J. Fadili, J. L. Starck, and F. Murtagh, "Inpainting and zooming using sparse representations," Comput. J. **52**, 64–79 (2007).
11. Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," IEEE Trans. Pattern Anal. Mach. Intell. **29**, 463–476 (2007).
12. T. Ruzic and A. Pizurica, "Context-aware patch-based image inpainting using Markov random field modeling," IEEE Trans. Image Process. **24**, 444–456 (2015).
13. K. Patwardhan, G. Sapiro, and M. Bertalmio, "Video inpainting under constrained camera motion," IEEE Trans. Image Process. **16**, 545–553 (2007).
14. C. Guillemot and O. Le Meur, "Image inpainting: overview and recent advances," IEEE Signal Process. Mag. **31**(1), 127–144 (2014).
15. M. K. Ng, H. Shen, E. Y. Lam, and L. Zhang, "A total variation regularization based super-resolution reconstruction algorithm for digital video," EURASIP J. Adv. Signal Process. **2007**, 1–6 (2007).
16. R. R. Sahay and A. N. Rajagopalan, "Joint image and depth completion in shape-from-focus: Taking a cue from parallax," J. Opt. Soc. Am. A **27**, 1203–1213 (2010).
17. A. V. Bhavsar and A. N. Rajagopalan, "Range map superresolution inpainting, and reconstruction from sparse data," Comput. Vis. Image Understanding **116**, 572–591 (2012).
18. Y. Zheng and C. Kambhamettu, "Learning based digital matting," in *IEEE Conference on Computer Vision* (IEEE, 2009), pp. 889–896.
19. G. Yu and J.-M. Morel, "ASIFT: An algorithm for fully affine invariant comparison," Image Proc. On Line **1**, 1–28 (2011).
20. T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," IEEE Trans. Pattern Anal. Mach. Intell. **33**, 500–513 (2011).
21. P. Felzenszwalb and D. Huttenlocher, "Efficient belief propagation for early vision," Int. J. Comput. Vis. **70**, 41–54 (2006).
22. Y. Mu, W. Liu, and S. Yan, "Video de-fencing," IEEE Trans. Circts. Sys. Vid. Tech. **24**, 1111–1121 (2014).
23. T. Xue, M. Rubinstein, C. Liu, and W. T. Freeman, "A computational approach for obstruction-free photography," ACM Trans. Graph. **34**, 1–11 (2015).
24. M. Park, K. Brocklehurst, R. Collins, and Y. Liu, "Deformed lattice detection in real-world images using mean-shift belief propagation," IEEE Trans. Pattern Anal. Mach. Intell. **31**, 1804–1816 (2009).
25. Y. Liu, R. Collins, and Y. Tsin, "A computational model for periodic pattern perception based on frieze and wallpaper groups," IEEE Trans. Pattern Anal. Mach. Intell. **26**, 354–371 (2004).
26. J. Hays, M. Leordeanu, A. Efros, and Y. Liu, "Discovering texture regularity as a higher-order correspondence problem," in *European Conference on Computer Vision* (2006), pp. 522–535.
27. W.-C. Lin and Y. Liu, "A lattice-based MRF model for dynamic near-regular texture tracking," IEEE Trans. Pattern Anal. Mach. Intell. **29**, 777–792 (2007).
28. A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," IEEE Trans. Pattern Anal. Mach. Intell. **30**, 228–242 (2008).
29. M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," in *IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2003), pp. 7–12.
30. A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," in *IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2003), Vol. **2**, pp. 721–728.
31. Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," IEEE Trans. Image Process. **19**, 1153–1165 (2010).
32. K. Papafitsoros, C. B. Schoenlieb, and B. Sengul, "Combined first and second order total variation inpainting using split Bregman," Image Process. On Line **3**, 112–136 (2013).
33. K. He and J. Sun, "Image completion approaches using the statistics of similar patches," IEEE Trans. Pattern Anal. Mach. Intell. **36**, 2423–2435 (2014).
34. M. Ebdelli, O. Le Meur, and C. Guillemot, "Video inpainting with short-term windows: application to object removal and error concealment," IEEE Trans. Image Process. **24**, 3034–3047 (2015).
35. J. Gu, R. Ramamoorthi, P. Belhumeur, and S. Nayar, "Removing image artifacts due to dirty camera lenses and thin occluders," ACM Trans. Graph. **28**, 1–144 (2009).
36. S. Bagon, Matlab Wrapper for Graph Cut, 2006, http://www.wisdom.weizmann.ac.il/~bagon.
37. Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," IEEE Trans. Pattern Anal. Mach. Intell. **23**, 1222–1239 (2001).
38. Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," IEEE Trans. Pattern Anal. Mach. Intell. **26**, 1124–1137 (2004).
39. V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" IEEE Trans. Pattern Anal. Mach. Intell. **26**, 147–159 (2004).
40. B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2010), pp. 2963–2970.
41. J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," J. Opt. Soc. Am. A **2**, 1160–1169 (1985).
42. J. Wang and M. Cohen, "Image and video matting: a survey," Found. Trends Comput. Graph. Vis. **3**, 97–175 (2008).
43. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE International Conference on Computer Vision* (IEEE, 2005), pp. 886–893.
44. C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, "Hoggles: visualizing object detection features," in *IEEE International Conference on Computer Vision* (IEEE, 2013), pp. 1–8.

45. C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," ACM Trans Intell. Syst. Technol. **2**, 1–27 (2011).
46. D. Sun, S. Roth, and M. Black, "Secrets of optical flow estimation and their principles," in *IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2010), pp. 2432–2439.
47. C. Liu, J. Yuen, and A. Torralba, "Sift flow: dense correspondence across scenes and its applications," IEEE Trans. Pattern Anal. Mach. Intell. **33**, 978–994 (2011).
48. P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: large displacement optical flow with deep matching," in *IEEE International Conference on Computer Vision* (IEEE, 2013), pp. 1385–1392.
49. R. Timofte and L. Van Gool, "Sparse flow: sparse matching for small to large displacement optical flow," in *IEEE Winter Conference on Applications of Computer Vision* (IEEE, 2015), pp. 1100–1106.
50. S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," in *IEEE International Conference on Computer Vision* (IEEE, 2007), pp. 1–8.
51. S. Z. Li, *Markov Random Field Modeling in Image Analysis* (Springer, 2001).
52. PSU NRT data set, http://vision.cse.psu.edu/data/MSBPLattice.shtml.
53. http://zoo.sandiegozoo.org/animals/giraffe (accessed September 30, 2015).
54. https://www.youtube.com/watch?v=6vMEtXqLh1A (accessed September 30, 2015).