Deep Learning Based Fence Segmentation and Removal from an Image Using a Video Sequence

Sankaraganesh Jonna[1](✉), Krishna K. Nakka[2], and Rajiv R. Sahay[2]

[1] Department of Computer Science and Engineering,
Indian Institute of Technology Kharagpur, Kharagpur, India
sankar9.iitkgp@gmail.com

[2] Department of Electrical Engineering, Indian Institute of Technology Kharagpur,
Kharagpur, India
krishkanth.92@gmail.com, sahayiitm@gmail.com

Abstract. Conventional approaches to image de-fencing use multiple adjacent frames for segmentation of fences in the reference image and are limited to restoring images of static scenes only. In this paper, we propose a de-fencing algorithm for images of dynamic scenes using an occlusion-aware optical flow method. We divide the problem of image de-fencing into the tasks of automated fence segmentation from a single image, motion estimation under known occlusions and fusion of data from multiple frames of a captured video of the scene. Specifically, we use a pre-trained convolutional neural network to segment fence pixels from a single image. The knowledge of spatial locations of fences is used to subsequently estimate optical flow in the occluded frames of the video for the final data fusion step. We cast the fence removal problem in an optimization framework by modeling the formation of the degraded observations. The inverse problem is solved using fast iterative shrinkage thresholding algorithm (FISTA). Experimental results show the effectiveness of proposed algorithm.

Keywords: Image inpainting · De-fencing · Deep learning · Convolutional neural networks · Optical flow

1 Introduction

Images containing fences/occlusions occur in several situations such as photographing statues in museums, animals in a zoo etc. Image de-fencing involves the removal of fences or occlusions in images. De-fencing a single photo is strictly an image inpainting problem which uses data in the regions neighbouring fence pixels in the frame for filling-in occlusions. The works of [1–4] addressed the image inpainting problem wherein a portion of the image which is to be inpainted is specified by a mask manually. As shown in Fig. 1(a), in the image de-fencing problem it is difficult to manually mark all fence pixels since they are numerous

G. Hua and H. Jégou (Eds.): ECCV 2016 Workshops, Part III, LNCS 9915, pp. 836–851, 2016.
DOI: 10.1007/978-3-319-49409-8_68

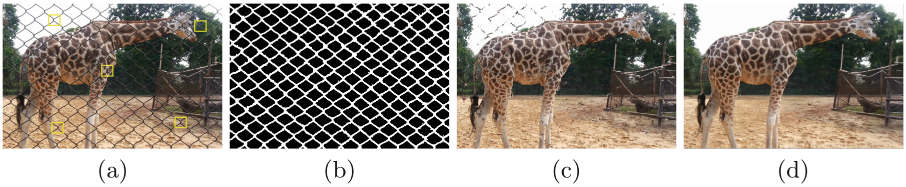


Fig. 1. (a) A frame taken from a video. (b) Segmented binary fence mask obtained using proposed CNN-SVM algorithm. (c) Inpainted image corresponding to (a) using the method of [2]. (d) De-fenced image corresponding to (a) using the proposed algorithm.

and spread over the entire image. The segmented binary fence mask obtained using the proposed algorithm is shown in Fig. 1(b). These masks are used in our work to aid in occlusion-aware optical flow computation and background image reconstruction. In Fig. 1(c), we show the inpainted image corresponding to Fig. 1(a) obtained using the method of [2]. The de-fenced image obtained using the proposed algorithm is shown in Fig. 1(d). As can be seen from Fig. 1(c), image inpainting does not yield satisfactory results when the image contains fine textured regions which have to be filled-in. However, using a video panned across a fenced scene can lead to better results due to availability of additional information in the adjacent frames.

Although, there has been significant progress in the area of lattice detection [5,6] and restoration of fenced images/videos [6–10], segmentation of fence or occlusion from a single image and de-fencing scenes containing dynamic elements are still challenging problems. Most of the existing works assume global motion between the frames and use images of static scene elements only [8–10]. Initial work related to image de-fencing has been reported by Liu et al. [7], wherein fence patterns are segmented via spatial regularity and the fence occlusions are filled-in using an inpainting algorithm [2]. Recent attempts for image de-fencing [9,10] use the parallax cue for fence pattern segmentation using multiple frames from a video. However, these works [9,10] constrain the scene elements to be static. Another drawback of [9] is that if the scene does not produce appreciable depth parallax fence segmentation is inaccurate. A very recent image de-fencing algorithm [6] exploits both color and motion cues for automatic fence segmentation from dynamic videos.

The proposed algorithm for image de-fencing uses a video captured by panning a camera relative to the scene and requires the solution of three subproblems. The first task is automatic segmentation of fence pixels in the frames of the captured video. Importantly, unlike existing works [6–10], we propose a machine learning algorithm to segment fences in a *single* image. We propose to use a pre-trained convolutional neural network (CNN) for fence texel joint detection to generate automatic scribbles which are fed to an image matting [11] technique to obtain the binary fence mask. Note that sample portions of images marked with yellow colored squares shown in Fig. 1(a) are treated as fence texels in this work. To the best of our knowledge, we are the first to detect fence texels using a pre-trained CNN coupled with an SVM classifier. Secondly, we estimate the pixel correspondence between the reference frame and the additional frames using a modified optical flow algorithm which incorporates the knowledge of location of occlusions in the observations. It is to be noted that existing optical flow algorithms find the relative shift only between pixels *visible* in two frames. Accurate registration of the observations is critical in de-fencing the reference image since erroneous pixel matching would lead to incorrect data fusion from additional frames. The basic premise of our work is that image regions occluded by fence pixels in the reference frame are rendered visible in other frames of the captured video. Therefore, we propose an occlusion-aware optical flow method using fence pixels located in the first step of our image de-fencing pipeline to accurately estimate background pixel correspondences even at occluded image regions. Finally, we fuse the information from additional frames in order to uncover the occluded pixels in the reference frame using an optimization framework. Since natural images are sparse, we use the fast iterative shrinkage thresholding algorithm (FISTA) to solve the resulting ill-posed inverse problem assuming l_1 norm of the de-fenced image as the regularization prior.

2 Prior Work

The problem of image de-fencing has been first addressed in [7] by inpainting fence pixels of the input image. The algorithm proposed in [12] used multiple images for de-fencing, which significantly improves the performance due to availability of occluded image data in additional frames. The work of [12] used a deformable lattice detection method proposed in [5] for fence detection. Unfortunately, the method of [5] is not a robust approach and fails for many real-world images. Khasare et al. [8] proposed an improved multi-frame de-fencing technique by using loopy belief propagation. However, there are two issues with their approach. Firstly, the work in [8] assumed that motion between the frames is global. This assumption is invalid for more complex dynamic scenes where the motion is non-global. Also, the method of [8] used an image matting technique proposed by [11] for fence segmentation which involves significant user interaction. A video de-fencing algorithm [9], proposed a soft fence segmentation method where visual parallax serves as the cue to distinguish fences from the unoccluded pixels. Recently, Xue et al. [10] jointly estimated the foreground masks and obstruction-free images using five frames taken from a video. Apart from the image based techniques, Jonna et al. [13] proposed a multimodal approach for image de-fencing wherein they have extracted the fence masks with the aid of depth maps corresponding to the color images obtained using the Kinect sensor. Very recently, our works [14,15] addresses the image de-fencing problem. However, the drawback of both the methods [14,15] is that they do not estimate occlusion-aware optical flow for data fusion.

The proposed algorithm for image de-fencing addresses some of the issues with the existing techniques. Firstly, we propose a machine learning algorithm using CNN-SVM for fence segmentation from a *single* image unlike existing works [6,9,10], which need a few frames to obtain the fence masks. Importantly, unlike the works of [9,10], the proposed algorithm does not assume that the scene is static but we can handle scenes containing dynamic elements. For this purpose, we propose a modified optical flow algorithm for estimation of pixel correspondence between the reference frame and additional frames after segmenting occlusions.

3 Methodology

We relate the occluded image to the original de-fenced image using a degradation model as follows,

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

where $\mathbf{y}_m$ are observations containing fences obtained from the captured video, $\mathbf{O}_m$ are the binary fence masks, $\mathbf{F}_m$ models the relative motion between frames, $\mathbf{x}$ is the de-fenced image and $\mathbf{n}_m$ is Gaussian noise. As described in Sect. 1, the problem of image de-fencing was divided into three sub-problems, which we elaborate upon in the following sub-sections.

3.1 Pre-trained CNN-SVM for Fence Texel Joint Detection

The important property of most outdoor fences is their symmetry about the fence texel joints. Referring to Fig. 1(a), we observe that fence texels appear repetitively throughout the entire image. Convolutional neural nets (CNN), originally proposed by [16], can be effectively trained to recognize objects directly from images with robustness to scale, rotation, translation, noise etc. Recently, Krizhevsky et al. [17] proved the utility of CNNs for object detection and classification in the ILSVRC challenge [18]. Since real-world fence texels exhibit variations in color, shape, noise, etc., we are motivated to use CNNs for segmenting these patterns robustly.

Convolutional neural networks belong to a class of deep learning techniques which operate directly on an input image extracting features using a cascade of convolutional, activation and pooling layers to finally predict the image category. The key layer in CNN is the convolutional layer whose filter kernels are learnt automatically via backpropagation. The commonly used non-linear activation functions are sigmoid, tanh, rectified linear unit (ReLU) and maxout [19] etc. The pooling layers sub-sample the input data. Overfitting occurs in neural networks when the training data is limited. Recently, a technique called Dropout [20] has been proposed which can improve the generalization capability of CNNs by randomly dropping some of the neurons.

However, since CNNs use supervised learning they need huge labeled datasets and long training time. A possible solution to this problem is to use transfer learning [21,22], wherein pre-trained models are used to initialize the weights and fine-tune the network on a different dataset. One can also preserve the pre-trained filter kernels and re-train the classifier part only. In this work, we used a CNN pre-trained on ImageNet [18] as a feature extractor by excluding the softmax layer.

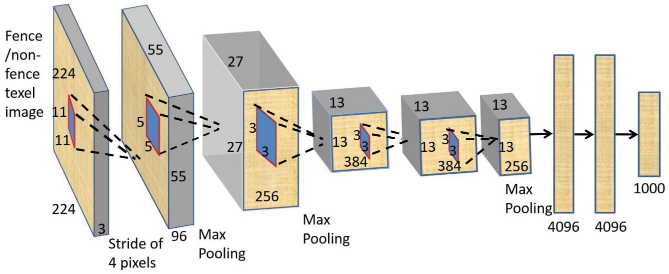
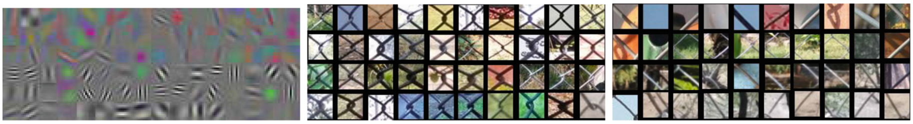


Fig. 2. The architecture of the pre-trained CNN [17].

The architecture of the CNN in Fig. 2 trained on ImageNet contains five convolutional layers followed by three fully-connected layers and a softmax classifier. Max-pooling layers follow first, second and fifth convolutional layer.

In Fig. 3(a), we show the 96 filter kernels of dimensions 11 × 11 × 3 learned by the first convolutional layer on input images. In this work, we propose to use CNN as a generic feature extractor followed by a support vector machine classifier (CNN-SVM). A given RGB input image is resized to $224 \times 224 \times 3$ and fed to the proposed CNN-SVM a feature vector of size 4096 is extracted from the seventh fully-connected layer.



(a) (b) (c)

Fig. 3. (a) 96 learned filter kernels of size 11 × 11 × 3 extracted from the first convolutional layer. (b) Sample fence texel joints. (c) Examples of non-fence texel joints.

An SVM classifier has been trained to detect fence texels using on these features of dimension 4096 extracted by the pre-trained CNN from a dataset of 20, 000 fence texel joints and 40, 000 non-fence texel sub-images. In Figs. 3(b) and (c), we show samples of fence texel texels and non-fence texels, respectively. During the testing phase, a sliding window is used to densely scan the test image shown in Fig. 4(a) from left to right and top to bottom with a stride of 5 pixels. The overall workflow of the proposed fence segmentation algorithm is shown in Fig. 4. Detected fence texels are joined by straight edges as shown in Fig. 4(b). In Fig. 4(c) we show the response obtained by Canny edge detection [23] algorithm after dilating the preliminary fence mask shown in Fig. 4(b) and treated as background scribbles. The combination of both foreground and background scribbles is shown in Fig. 4(d), wherein foreground scribbles are obtained by erosion operation on the image in Fig. 4(b). We fed these automatically generated

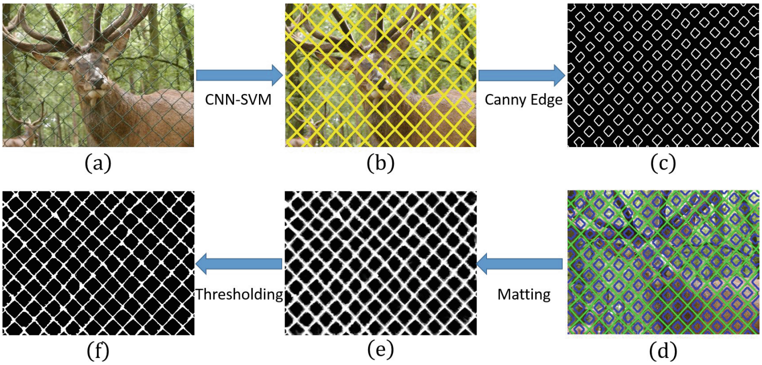


Fig. 4. Schematic of fence mask segmentation.

scribbles to the method of [11] and obtain the alpha map in Fig. 4(e). Finally, the binary fence mask shown in Fig. 4(f) is generated by thresholding the alpha map obtained from [11].

3.2 Occlusion Aware Optical Flow

The image alignment problem becomes more complex when real-world videos contain dynamic objects. Handling motion boundaries and occlusions in videos for optical flow computation is still challenging. Internal occlusions due to the layered dynamic objects and external occlusions such as fences make the problem tougher. In some practical applications of computer vision such as view synthesis, image de-fencing, etc. we need to compute the correspondence of all pixels between two images despite occlusions. Many algorithms for estimating optical flow are proposed in the literature [24–27], which are based on modifications of the basic variational framework proposed by Horn et al. [28] addressing its various shortcomings. Recently, significant progress has been made in order to compute dense optical flow in a robust manner [25,29,30]. The state-of-the-art optical flow algorithms [24,25] integrate descriptor matching between two images in a variational framework. It is due to a robust function in the variational framework that the algorithm in [24] can handle small internal occlusions. However, it fails to tackle large external occlusions. The algorithm of [29] computes dense correspondence between images by performing sparse-dense interpolation under contour and motion boundary assumption. An occlusion aware optical flow algorithm is proposed by [31], wherein occlusions in images are handled using a three-step procedure. Initially, the method in [31] estimates occlusion-ignorant optical flow. Subsequently, occlusions are computed using this unreliable optical flow. Finally, the algorithm in [31] corrects the optical flow using estimated occlusions.

The basic cue behind the proposed image de-fencing algorithm is that occluded image data in the reference frame is uncovered in additional frames of the captured video. Relative motion among observations needs to be estimated to fuse the information uncovered in the additional images for filling in occlusions in the reference frame. State-of-the-art optical flow algorithms estimate the flow of *visible* areas between two images. However, as described above, there are occlusions in images due to depth changes, dynamic scene elements and external hindrances such as fences/barricades. If we apply the conventional optical flow algorithms to register two images containing fence occlusions we encounter two difficulties while aligning corresponding fence and background pixels. Firstly, large motion discontinuities exist at the spatial location of fences due to abrupt depth changes which corrupt the estimated optical flow. Secondly, it is to be noted that the background pixels hidden behind the fence assume the flow of fence pixels instead of their own ground truth motion. Hence, in this work we modify the motion associated with fence pixels to that of surrounding background pixel motion in order to reveal the occluded pixel information in the warped adjacent frame.

In this paper, we re-formulate the optical flow algorithm of [32] to fit our application of image de-fencing. Akin to [32], coarse to fine optical flow is estimated using an incremental framework in Gaussian scale-space. Note that we have already obtained the binary fence mask $\mathbf{O}_m$ corresponding to the segmented fence pixels in the observation $\mathbf{y}_m$. We insert this mask $\mathbf{O}_m$ as occlusion operator inside the optical flow framework to deal with the motion inaccuracies at fence locations. At the fence locations data cost is assumed to be zero and only smoothness term in Eq. (3) guides optical flow estimation. We assume sparse gradient prior (modeled using l_1 norm) for both horizontal and vertical velocities. At every scale, the optimized values are up-scaled and used as initial estimate at the next fine scale.

Suppose $\mathbf{w} = [u, v]$ be the current estimate of horizontal and vertical flow fields and $\tilde{y}_r, \tilde{y}_t$ be the reference and t^{th} adjacent images, respectively. Under the incremental framework [32,33], one needs to estimate the best increment $d\mathbf{w} = (du, dv)$ as follows

$$E(du, dv) = \arg\min_{d\mathbf{w}} \parallel \mathbf{F}_{\mathbf{w}+d\mathbf{w}}\tilde{y}_t - \tilde{y}_r \parallel_1 + \mu \parallel \nabla(u + du) \parallel_1 + \mu \parallel \nabla(v + dv) \parallel_1 \tag{2}$$

where $\mathbf{F}_{\mathbf{w}+d\mathbf{w}}$ is the warping matrix corresponding to flow $\mathbf{w} + d\mathbf{w}$, ∇ is the gradient operator and μ is the regularization parameter. To use gradient based methods, we replace the l_1 norm with a differentiable approximation $\phi(x^2) = \sqrt{x^2 + \epsilon^2}$, $\epsilon = 0.001$. To robustly estimate optical flow under the known fence occlusions we compute the combined binary mask $\mathbf{O} = \mathbf{F}_{\mathbf{w}+d\mathbf{w}}\mathbf{O}_t || \mathbf{O}_r$ obtained by the logical OR operation between the reference fence mask and backwarped fence from the t^{th} frame using warping matrix $\mathbf{F}_{\mathbf{w}+d\mathbf{w}}$. To estimate the optical flow increment in the presence of occlusions we disable the data fidelity term by incorporating $\mathbf{O}$ in Eq. (2) as

$$E(du, dv) = \arg\min_{d\mathbf{w}} \parallel \mathbf{O}(\mathbf{F}_{\mathbf{w}+d\mathbf{w}}\tilde{y}_t - \tilde{y}_r) \parallel_1 + \mu \parallel \nabla(u + du) \parallel_1 + \mu \parallel \nabla(v + dv) \parallel_1 \tag{3}$$

By first-order Taylor series expansion,

$$\mathbf{F}_{\mathbf{w}+d\mathbf{w}}\tilde{y}_t \approx \mathbf{F}_{\mathbf{w}}\tilde{y}_t + \mathbf{Y}_x du + \mathbf{Y}_y dv \tag{4}$$

where $\mathbf{Y}_x = diag(\mathbf{F}_{\mathbf{w}}\tilde{y}_{t_x})$, $\mathbf{Y}_y = diag(\mathbf{F}_{\mathbf{w}}\tilde{y}_{t_y})$, $\tilde{y}_{t_x} = \frac{\partial}{\partial x}\tilde{y}_t$ and $\tilde{y}_{t_y} = \frac{\partial}{\partial y}\tilde{y}_t$. We can write Eq. (3) as

$$\begin{aligned}\arg\min_{d\mathbf{w}} \parallel \mathbf{O}\mathbf{F}_{\mathbf{w}}\tilde{y}_t + \mathbf{O}\mathbf{Y}_x du + \mathbf{O}\mathbf{Y}_y dv - \mathbf{O}\tilde{y}_r) \parallel_1 &+\mu \parallel \nabla(u+du) \parallel_1 \\ &+\mu \parallel \nabla(v+dv) \parallel_1\end{aligned} \tag{5}$$

To estimate the best increments du, dv to the current flow u, v we equate the gradients $\left[\frac{\partial E}{\partial du}; \frac{\partial E}{\partial dv}\right]$ to zero.

$$\begin{aligned}&\begin{bmatrix}\mathbf{Y}_x^T\mathbf{O}^T\mathbf{W}_d\mathbf{O}\mathbf{Y}_x + \mu L & \mathbf{Y}_x^T\mathbf{O}^T\mathbf{W}_d\mathbf{O}\mathbf{Y}_y \\ \mathbf{Y}_y^T\mathbf{O}^T\mathbf{W}_d\mathbf{O}\mathbf{Y}_x & \mathbf{Y}_y^T\mathbf{O}^T\mathbf{W}_d\mathbf{O}\mathbf{Y}_y + \mu L\end{bmatrix}\begin{bmatrix}du \\ dv\end{bmatrix} \\ &= \begin{bmatrix}-Lu - \mathbf{Y}_x^T\mathbf{O}^T\mathbf{W}_d\mathbf{O}\mathbf{F}_{\mathbf{w}}\tilde{y}_t + \mathbf{Y}_x^T\mathbf{O}^T\mathbf{W}_d\mathbf{O}\tilde{y}_r \\ -Lv - \mathbf{Y}_y^T\mathbf{O}^T\mathbf{W}_d\mathbf{O}\mathbf{F}_{\mathbf{w}}\tilde{y}_t + \mathbf{Y}_y^T\mathbf{O}^T\mathbf{W}_d\mathbf{O}\tilde{y}_r\end{bmatrix}\end{aligned}$$

where $L = \mathbf{D}_x^T\mathbf{W}_s\mathbf{D}_x + \mathbf{D}_y^T\mathbf{W}_s\mathbf{D}_y$, $\mathbf{W}_s = diag(\phi'(|\nabla u|^2))$ and $\mathbf{W}_d = diag(\phi'(|\mathbf{O}\mathbf{F}_{\mathbf{w}}\tilde{y}_t - \mathbf{O}\tilde{y}_r|^2))$. We define $\mathbf{D}_x$ and $\mathbf{D}_y$ are discrete differentiable operators along horizontal and vertical directions, respectively. We used conjugate gradient (CG) algorithm to solve for $d\mathbf{w}$ using iterative re-weighted least squares (IRLS) framework.

3.3 FISTA Optimization Framework

Once the relative motion between the frames has been estimated we need to fill-in the occluded pixels in the reference image using the corresponding uncovered pixels from the additional frames. Reconstructing de-fenced image $\mathbf{x}$ from the occluded observations is an ill-posed inverse problem and therefore prior information for $\mathbf{x}$ has to be used to regularize the solution. Since natural images are sparse, we employed l_1 norm of the de-fenced image as regularization constraint in the optimization framework as follows,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left[\sum_m \parallel \mathbf{y}_m^{obs} - \mathbf{O}_m\mathbf{F}_m\mathbf{x} \parallel^2 + \lambda \parallel \mathbf{x} \parallel_1\right] \tag{6}$$

where λ is the regularization parameter.

Since the objective function contains l_1 norm as a regularization function, it is difficult to solve Eq. 6 with the conventional gradient-based algorithms. Here, we employed one of the proximal algorithms such as FISTA [34] iterative framework to handle non-smooth functions for image de-fencing. The key step in FISTA iterative framework is the proximal operator [35] which operates on the combination of two previous iterates.

Algorithm 1. FISTA image de-fencing

```
Input: $\lambda, \alpha, \mathbf{z}_1 = \mathbf{x}_0 \in \mathbb{R}^{M \times N}, t_1 = 1$
repeat
    $\mathbf{x}_k = prox_\alpha(g)(\mathbf{z}_k - \alpha \nabla f(\mathbf{z}_k))$
    $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
    $\mathbf{z}_{k+1} = \mathbf{x}_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(\mathbf{x}_k - \mathbf{x}_{k-1})$
    $k \leftarrow k + 1$
until $(\| \mathbf{x}_k - \mathbf{x}_{k-1} \|_2 \leq \epsilon)$
```

The proximal operator is defined as the solution of the following convex optimization [36]

$$prox_\alpha(g)(x) = \arg\min_y \{g(y) + \frac{1}{2\alpha} \| y - x \|^2\} \tag{7}$$

If $g(y)$ is l_1 norm, then $prox_\alpha(g)(x) = max(|x| - \lambda\alpha, 0)sign(x)$. The gradient for data matching cost f is given as follows

$$\nabla f(\mathbf{z}) = \sum_m \mathbf{F}_m^T \mathbf{O}_m^T (\mathbf{O}_m \mathbf{F}_m \mathbf{z} - \mathbf{y}_m^{obs}) \tag{8}$$

4 Experimental Results

Initially, we report both qualitative and quantitative results obtained using the proposed fence segmentation algorithm on various datasets. Subsequently, we show the impact of accounting for occlusions in the incremental flow framework. Finally, we report image de-fencing results obtained with the FISTA optimization framework. To demonstrate the efficacy of the proposed de-fencing system, we show comparison results with state-of-the-art fence segmentation, and de-fencing methods in the literature. We used only three frames from each captured video for all the image de-fencing results reported here using the proposed algorithm. For all our experiments, we fixed $\lambda = 0.0005$ in Eq. 6. We ran all our experiments on a 3.4 GHz Intel Core i7 processor with 16 GB of RAM.

4.1 Fence Segmentation

For validating the proposed algorithm for fence segmentation, we have evaluated our algorithm on state-of-the-art datasets [9,10,37]. We also show segmentation results on a proposed fenced image dataset consisting of 200 real-world images captured under diverse scenarios and complex backgrounds. We report quantitative results on PSU NRT [37] dataset and qualitative results on [9,10,37] datasets. As discussed in Sect. 3.1, we have extracted features from $20,000$ fence, $40,000$ non-fence texel images using a pre-trained CNN to train an SVM classifier. The trained classifier is used to detect joint locations in images via a sliding window protocol. We compare the results obtained using a state-of-the-art lattice detection algorithm [5] and the proposed algorithm on all the datasets.

Initially, in Fig. 5(a) we show a fenced image from the PSU NRT dataset [37]. Fence texels are detected using our pre-trained CNN-SVM approach and are jointed by straight edges, as shown in Fig. 5(f). Note that all fence texels are detected accurately in Fig. 5(f). In contrast, the method of [5] failed completely to extract the fence pixels as seen in Fig. 5(k). The output of Fig. 5(f) is used to generate foreground and background scribbles which are fed to the image matting technique of [11]. The final binary fence mask obtained by thresholding the output of [11] is shown in Fig. 5(p). Next, we have validated both the algorithms on image taken from a recent dataset [10] shown in Fig. 5(b). In Fig. 5(g), we show the fence texels detected using our pre-trained CNN-SVM approach and joined by straight edges. In contrast, the method of [5] failed completely to extract the fence pixels as seen in Fig. 5(l). The output of Fig. 5(g) is used to generate scribbles as outlined in Sect. 3.1. These foreground and background scribbles are fed to the image matting technique of [11]. The final binary fence mask obtained by thresholding the output of [11] is shown in Fig. 5(q). Finally, we perform experiments on images from the proposed fenced image dataset. Sample images taken from the dataset are shown in Figs. 5(c)–(e). In Figs. 5(h)–(j), we show the fence segmentations obtained using the proposed pre-trained CNN-SVM algorithm. We observe that the proposed algorithm detected all the fence texel joints accurately. The lattice detected using [5] are shown in Figs. 5(m)–(o). We can observe that the approach of [5] partially segments the fence pixels in Fig. 5(m). Note that in Fig. 5(o) the algorithm of [5] completely failed to segment fence pixels. The final binary fence masks obtained by thresholding the output of [11] are shown in Figs. 5(r)–(t).

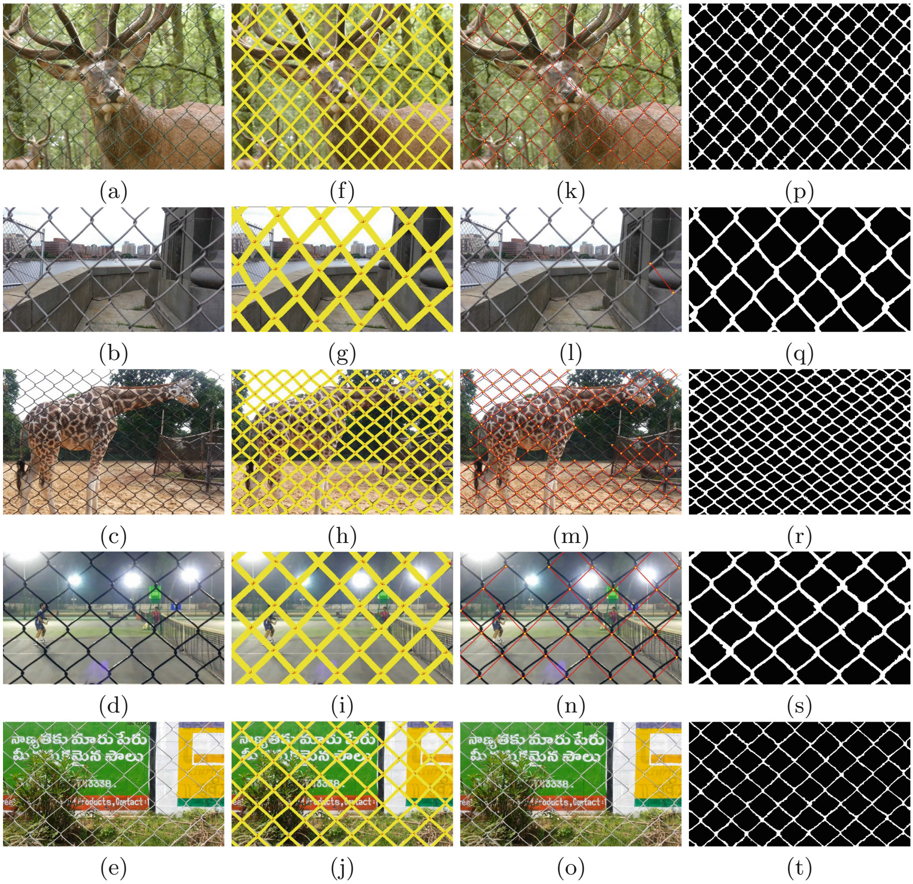
A summary of the quantitative evaluation of the fence texel detection method of [5] and the pre-trained CNN-SVM based proposed algorithm is given in Table 1. The F-measure obtained for [5] on PSU NRT [37] dataset and proposed fenced image datasets are 0.62 and 0.41, respectively. In contrast, F-measure for the proposed method on PSU NRT dataset [37] and our fenced image datasets are 0.97 and 0.94, respectively.

Table 1. Quantitative evaluation of fence segmentation

	NRT database [37]			Our database		
Method	Precision	Recall	F-measure	Precision	Recall	F-measure
Park et al. [5]	0.95	0.46	0.62	0.94	0.26	0.41
pre-trained CNN-SVM	0.96	0.98	**0.97**	0.90	0.98	**0.94**

4.2 Optical Flow Under Known Occlusions

To demonstrate the robustness of proposed optical flow algorithm under known occlusions, we use frames from videos of fenced scenes in [6,9,10]. We show two frames from a video sequence named "football" from [9] in the first column



(a) (f) (k) (p)

(b) (g) (l) (q)

(c) (h) (m) (r)

(d) (i) (n) (s)

(e) (j) (o) (t)

Fig. 5. First column: sample images from NRT [10,37] and proposed fenced image datasets, respectively. Second column: fence masks generated using the proposed pre-trained CNN-SVM algorithm. Third column: fence detection using [5]. Fourth column: final binary fence masks corresponding to images in the first column obtained by generating scribbles using fence detections in images of the second column which are fed to the method of [11].

of Fig. 6. The video sequences named “fence1” and “fence4” are taken from the work of [10]. Two frames from each of these videos are shown in second and third columns of Fig. 6, respectively. Video sequences named “lion” and “walking” are taken from [6] and a couple of observations from each of them are depicted in fourth and fifth columns of Fig. 6, respectively. In the third row of Fig. 6, we show the color coded optical flows obtained using [24] between respective images shown in each column of first and second row of Fig. 6. Note that the images shown in third row of Fig. 6 contain regions of erroneously estimated optical flow due to fence occlusions. In contrast, the flow estimated using proposed algorithm

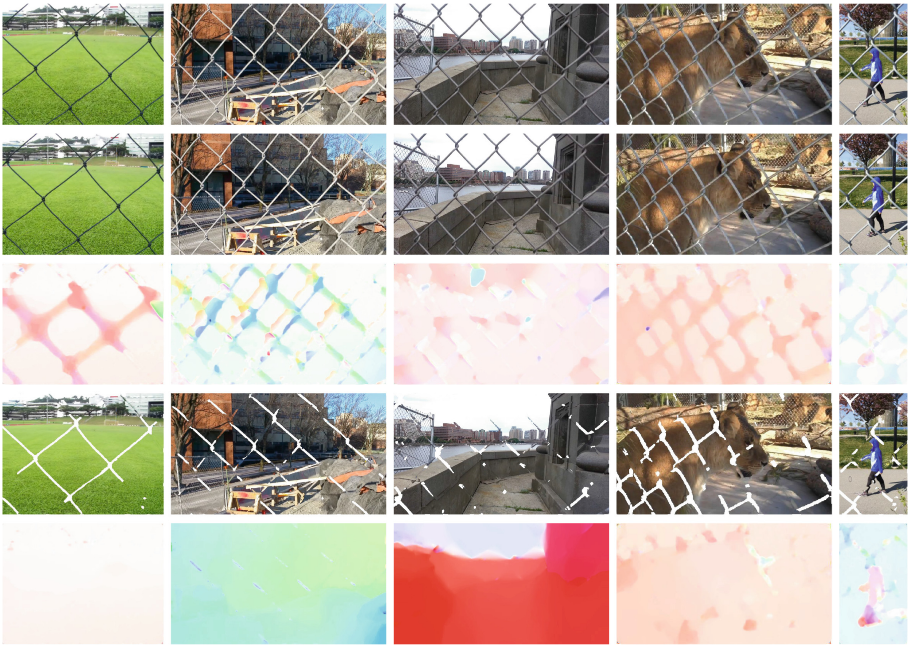
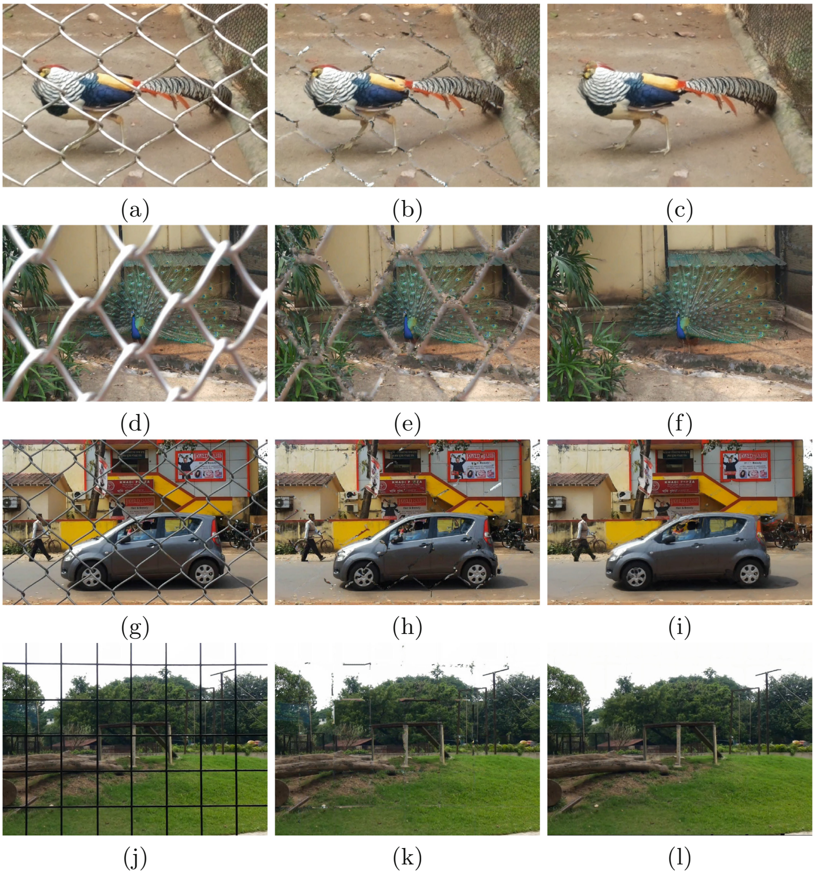


Fig. 6. First and second row: frames taken from videos reported in [6,9,10]. Third row: optical flow computed between the first and second row images using [24]. Fourth row: de-fenced images obtained using the estimated flow shown in the third row. Fifth row: occlusion-aware optical flow obtained using the proposed algorithm.

under known fence occlusions are shown in the fifth row of Fig. 6. Note that the optical flows estimated using the proposed method contain no artifacts.

4.3 Image De-Fencing

To demonstrate the efficacy of the proposed image de-fencing algorithm, we conducted experiments with several real-world video sequences containing dynamic background objects. In Figs. 7(a), (d), (g), and (j), we show the images taken from four different video sequences. The fence pixels corresponding to these observations are segmented using the proposed pre-trained CNN-SVM and the approach of [11]. In Figs. 7(b), (e), (h), and (k), we show the inpainted images obtained using [2] which was the method used for obtaining the de-fenced image after fence segmentation in [6]. Note that we can see several artifacts in the inpainted images obtained using [2]. De-fenced images obtained using the proposed algorithm are shown in Figs. 7(c), (f), (i), and (l), respectively. We observe that the proposed algorithm has effectively reconstructed image data even for dynamic real-world video sequences. Also, note that for all the results shown in Figs. 7(c), (f), (i), and (l) we used only three observations from the captured video sequences.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 7. First column: one frame each taken from challenging real-world videos. Second column: inpainted images obtained using exemplar-based image inpainting algorithm [2] which was the approach used in [6] for image de-fencing. Third column: de-fenced images obtained using the proposed algorithm corresponding to images in the first column.

Next, we compare the proposed algorithm with recent state-of-the-art methods [6,9,10]. In Fig. 8(a), we show the de-fenced image obtained using [9]. The corresponding result obtained by the proposed algorithm is shown in Fig. 8(e). Note that the de-fenced image obtained in [9] is blurred whereas the proposed algorithm generated a sharper image. We show a cropped region from both Figs. 8(a) and (e) in the last row to confirm our observation. In Figs. 8(b) and (f), we show the de-fenced results obtained by [10] and the proposed algorithm, respectively. The de-fenced image obtained using the method in [10] is distorted at some places which is apparent in Fig. 8(b). In contrast, the fence has been removed completely with hardly any distortions in the result shown in Fig. 8(f), which has been obtained using our algorithm. A cropped region from both Figs. 8(b) and (f) are shown in the last row to prove our point. The de-fenced images obtained using a very recent technique [6] are shown in Figs. 8(c) and (d), respectively. These results contain several artifacts. However, the de-fenced images recovered using the proposed algorithm hardly contain any artifacts as shown in Figs. 8(g) and (h). A cropped regions from Figs. 8(c) and (d) and Figs. 8(g) and (h) are shown in the last row for comparison purpose. Since we use only three frames from the videos, our method is more computationally efficient than [9,10] which use 5 and 15 frames, respectively.

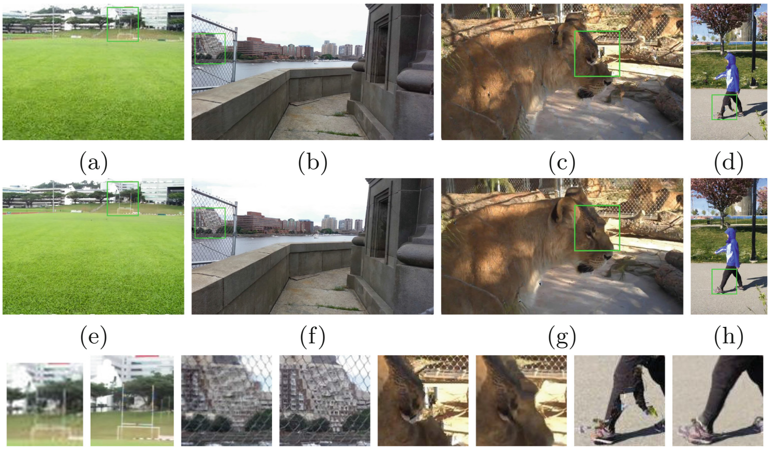


Fig. 8. Comparison with state-of-the-art image/video de-fencing methods [6,9,10] using video sequences from their works. (a) De-fenced image obtained by [9]. (b) Recovered background image using [10]. (c), (d) Inpainted images obtained by [2] which was the method used in [6]. (e)–(h) De-fenced images obtained by the proposed algorithm using occlusion-aware-optical flow shown in fifth row of Fig. 6. Last row: Insets from the images of first and second rows, respectively, showing the superior reconstruction of the de-fenced image by our algorithm.

5 Conclusions

In this paper, we proposed an automatic image de-fencing system for real-world videos. We divided the problem of image de-fencing into three tasks and proposed an automatic approach for each one of them. We formulated an optimization framework and solved the inverse problem using the fast iterative shrinkage thresholding algorithm (FISTA) assuming l_1 norm of the de-fenced image as the regularization constraint. We have evaluated the proposed algorithm on various datasets and reported both qualitative and quantitative results. The obtained results show the effectiveness of proposed algorithm. As part of future work, we are investigating how to optimally choose the frames from the video for fence removal.

References

1. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proceedings of ACM SIGGRAPH, pp. 417–424 (2000)
2. 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)
3. Hays, J., Efros, A.A.: Scene completion using millions of photographs. ACM Trans. Graph. **26**(3), 1–7 (2007)
4. Papafitsoros, K., Schonlieb, C.B., Sengul, B.: Combined first and second order total variation inpainting using split bregman. Image Process. On Line **3**, 112–136 (2013)
5. Park, M., Brocklehurst, K., Collins, R., Liu, Y.: Deformed lattice detection in real-world images using mean-shift belief propagation. IEEE Trans. Pattern Anal. Mach. Intell. **31**, 1804–1816 (2009)
6. Yi, R., Wang, J., Tan, P.: Automatic fence segmentation in videos of dynamic scenes. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) June 2016
7. Liu, Y., Belkina, T., Hays, J., Lublinerman, R.: Image de-fencing. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
8. Khasare, V.S., Sahay, R.R., Kankanhalli, M.S.: Seeing through the fence: Image de-fencing using a video sequence (2013)
9. Mu, Y., Liu, W., Yan, S.: Video de-fencing. IEEE Trans. Circts. Syst. Video Technol. **24**(7), 1111–1121 (2014)
10. Xue, T., Rubinstein, M., Liu, C., Freeman, W.T.: A computational approach for obstruction-free photography. ACM Trans. Graph. **34**(4), Article no. 79 (2015)
11. Zheng, Y., Kambhamettu, C.: Learning based digital matting. In: International Conference on Computer Vision (ICCV) (2009)
12. Park, M., Brocklehurst, K., Collins, R.T., Liu, Y.: Image de-fencing revisited (2010)
13. Jonna, S., Voleti, V.S., Sahay, R.R., Kankanhalli, M.S.: A multimodal approach for image de-fencing and depth in painting. In: Proceedings of International Conference on Advances in Pattern Recognition, pp. 1–6 (2015)
14. Jonna, S., Nakka, K.K., Sahay, R.R.: My camera can see through fences: a deep learning approach for image de-fencing. In: 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), pp. 261–265, November 2015
15. 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. A **33**(10), 1917–1930 (2016)
16. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. **25**, 1097–1105 (2012)
18. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR09 (2009)
19. Goodfellow, I.J., Warde-farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. In: ICML (2013)
20. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
21. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: a deep convolutional activation feature for generic visual recognition. CoRR abs/1310.1531 (2013)

22. Vedaldi, A., Lenc, K.: Matconvnet - convolutional neural networks for MATLAB. CoRR abs/1412.4564 (2014)
23. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. PAMI **8**(6), 679–698 (1986)
24. Brox, T., Malik, J.: Large displacement optical flow: descriptor matching in variational motion estimation. IEEE Trans. Pattern Anal. Mach. Intell. **33**(3), 500–513 (2011)
25. Xu, L., Jia, J., Matsushita, Y.: Motion detail preserving optical flow estimation. IEEE Trans. Pattern Anal. Mach. Intell. **34**(9), 1744–1757 (2012)
26. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24673-2_3
27. Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T.: SIFT Flow: dense correspondence across different scenes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5304, pp. 28–42. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88690-7_3
28. Horn, B.K., Schunck, B.G.: Determining optical flow. Technical report, Cambridge, MA, USA (1980)
29. Revaud, J., Weinzaepfel, P., Harchaoui, Z., Schmid, C.: EpicFlow: edge-preserving interpolation of correspondences for optical flow. In: Computer Vision and Pattern Recognition (2015)
30. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: Deepflow: large displacement optical flow with deep matching. In: Proceedings of International Conference on Compututer Vision, pp. 1385–1392, December 2013
31. Ince, S., Konrad, J.: Occlusion-aware optical flow estimation. IEEE Trans. Image Process. **17**(8), 1443–1451 (2008)
32. Liu, C.: Beyond pixels: exploring new representations and applications for motion analysis. Ph.D. thesis, Massachusetts Institute of Technology (2009)
33. Liu, C., Sun, D.: On Bayesian adaptive video super resolution. IEEE Trans. Pattern Anal. Mach. Intell. **36**(2), 346–360 (2014)
34. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imag. Sci. **2**(1), 183–202 (2009)
35. Barbero, A., Sra, S.: Fast Newton-type methods for total variation regularization. In: ICML, pp. 313–320. Omnipress (2011)
36. Parikh, N., Boyd, S.: Proximal algorithms. Found. Trends Optim. **1**(3), 123–231 (2014)
37. PSU NRT data set: http://vision.cse.psu.edu/data/MSBPLattice.shtml