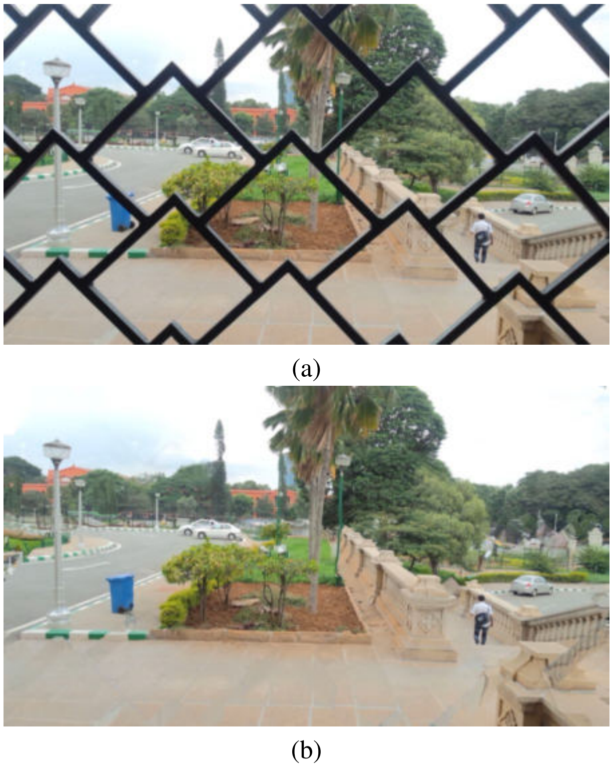
An Approach for Image Dis-occlusion and Depth Map Completion using Computational Cameras

Sankaraganesh Jonna, *Member, IEEE,* Sukla Satapathy, *Student Member, IEEE,* Vikram S. Voleti, *Student Member, IEEE,* and Rajiv R. Sahay, *Member, IEEE*

***Abstract*—Low-cost computational cameras have enabled the use of depth channel information along with color images, and the same will be possible with next-generation smartphones. However, occlusions in the scene pose challenges for the amateur photographer. This paper addresses the problem of automatic removal of fences/occlusions from an image using the video of a static/dynamic scene. We also perform depth completion by fusing data from multiple recorded depth maps affected by occlusions or holes. Interestingly, in this work we harness *depth cue* for fence segmentation. However, accurate estimation of the relative shifts between captured color frames and depth maps of the RGB-D video data is necessary. To handle this challenging task, we propose a robust algorithm to estimate optical flow under known fences/occlusions. In order to preserve discontinuities in the de-fenced image/inpainted depth map, we formulate an optimization framework using the total variation of the unoccluded image and the completed depth map as regularization constraints. To preserve discontinuities along the boundaries of depth layers, we also integrate a non-local regularization term along with the local total variation prior on the depth map. The ill-posed inverse problem of simultaneous estimation of the de-fenced image and the inpainted depth map is solved using the split Bregman iterative algorithm. To demonstrate the effectiveness of the proposed approach, we have conducted several experiments on real-world videos containing both static and dynamic scene elements captured using Kinect, Lytro and smartphone cameras.**

***Index Terms*—Inpainting, RGB-D data, light field, Lytro, optical flow, non-local means, total variation, convolutional neural networks.**



(a)

(b)

Fig. 1. (a) RGB image captured using smartphone camera which is obstructed by a fence occlusion. (b) Occlusion-free image corresponding to (a).

I. Introduction

In photography, sometimes one cannot avoid taking photos/videos through obstructions; for example, while capturing the video of a moving animal behind a fence in a zoo, or recording an activity through a grilled window. Particularly, visitors to tourist destinations often feel hindered in capturing photographs/videos of objects that are occluded by barricades/fences for security purposes. Those images/videos are unpleasant to see, and limit the performance of recognition systems due to degradation caused by occlusions. We show such an image in Fig. 1 (a), wherein the occluding background scene appears unpleasant to the user. In Fig. 1 (b), we present the obstruction-free image recovered using our proposed algorithm. In general, it is very challenging to restore occluded images/videos of dynamic scene elements automatically by changing camera aperture settings, using photo-editing software, or even with state-of-the-art occlusion removal algorithms [1–5]. To preserve the aesthetic appeal and fine details of the captured videos/images, one needs a more robust and sensitive algorithm to remove such fences/occlusions.

Although several image de-fencing algorithms have appeared recently in literature [1–8], automatic segmentation of fences/occlusions and their removal from videos of static/dynamic scenes is still a challenging problem. Moreover, fences exhibit complex shapes, arbitrary colors and varying widths, which increase the complexity of the problem. Two state-of-the-art image de-fencing algorithms [2, 3] used the parallax cue for foreground segmentation, and work well for videos of static scenes. Another recently proposed automatic fence segmentation algorithm [4] extracts foreground fence pixels from a dynamic video, and uses the image inpainting technique of [9] to fill in the occluded pixels. However, the main disadvantage of the algorithm in [4] is that it does not exploit the *temporal cue* for data fusion using temporal

M. Shell was with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: (see http://www.michaelshell.org/contact.html).

J. Doe and J. Doe are with Anonymous University.

Manuscript received April 19, 2005; revised August 26, 2015.

information in adjacent frames of the video. We show that mere inpainting of the occluded frames will not provide satisfactory results if the background scene is highly detailed.

We observe the fact that fences/occlusions in the scene are almost always closer to the camera/sensor than the subjects of the scene. If one could access the geometric description of the dynamic scene, it could lead to robust separation of foreground occlusions from the background scene. Therefore, in this paper, we harness *depth cue* to automatically segment fences or occluded regions in an image. Depth maps can be obtained using conventional passive shape estimation algorithms such as a stereo or active sensing technologies, or alternatively by using light-fields. Traditional passive algorithms compute the depth of a scene using two or multiple views by estimating the correspondences among the frames without active illumination. However, they demand exact image rectification, and produce erroneous estimates at texture-less regions. This limits their utility in many practical computer vision applications [10]. Active depth estimation techniques such as laser scanners, structured-light and time-of-flight (ToF) based techniques are used by many researchers in recent times. Among all existing techniques, state-of-the-art laser scanners produce depth maps of the highest quality and resolution. However, one cannot use laser scanners for real-time photo editing or graphics applications, since they are very expensive and extremely slow.

Structured-light cameras like the Kinect work on the principle of active triangulation. They emit a specific pattern (eg., striped, continuous texture or random dots) towards the scene, and an IR camera computes the depth map based on deformation of the pattern. Such cameras address the critical issue of computing reliable correspondences at texture-less regions using stereo algorithms. Examples of consumer-grade depth sensors are Microsoft Kinect v1, Intel RealSense F200, and Intel RealSense R200. Microsoft Kinect has brought low-cost active sensing technology (i.e Kinect v1 and v2) to the common masses [11] with significant depth resolution. Moreover, the spatial resolution of depth maps captured by these cameras is 640×480 pixels. Due to the offset between the two cameras in stereo and structured light systems, missing depth values near the occlusions cannot be avoided. The light source in ToF cameras emits a near-infrared light which is then reflected by the objects in the scene. Depth can be measured by computing the phase difference between the emitted and reflected light. For example commercial ToF sensors such as PMD, Swiss Ranger, and Kinect v2 record range maps of spatial resolutions 64×48, 176×144, and 512×424 pixels, respectively [12]. As per our knowledge, among all the consumer-grade TOF depth cameras, Kinect v2 measures depth at the highest resolution. In contrast to structured light systems, TOF systems have collinear transmitter and receiver, and produce more reliable depth estimates.

In recent years, light field imaging has become popular with the availability of commercial light field cameras such as Lytro [13] and Raytrix [14]. A ight field can be considered as an array of images captured in a single shot by a grid of cameras (microlens array) looking towards the scene. It can be represented as a $4D$ function $L(x, y, u, v)$, where x,y and u,v correspond to the spatial and angular dimensions respectively. Conventional cameras lose a lot of information about the $3D$ scene, whereas $4D$ light field data provides richer description (eg., refocusing, extended depth-of-field and multi-view) to enable a wide range of applications in computer vision, like depth estimation [15, 16], saliency detection [17], matting [18], and segmentation [19], etc.

To illustrate our assertions, in Figs. 2 (b), (d), depth maps corresponding to RGB images in (a) and (c) captured using Kinect v1 and v2 sensors, respectively, are shown. We notice that the depth map obtained using Kinect v1 is sensitive to occlusions and is affected by artifacts due to shadows, whereas that of Kinect v2 is more robust. The other advantage of Kinect is that we can capture depth data along with RGB images in real-time with free hand motion of the sensor. An image and its corresponding depth taken from the Stanford light field occlusion dataset [22] are shown in Figs. 2 (e) and (f) respectively. In Figs. 2 (g) and (h), we show a color image taken from the dataset in [4] and the corresponding disparity map obtained from it using the recently proposed stereo algorithm in [20]. We observe that passive algorithms such as stereo produce depth maps which suffer in quality around depth discontinuities and texture-less regions. Finally, the RGB image and corresponding depth image captured using a laser scanner are shown in Fig. 2 (i) and (j), obtained from [23].

Active sensing devices such as Kinect v1 [21] and v2 can capture RGB-D data in real-time with sufficiently high quality, and are inexpensive as well. Moreover, the quality of depth map obtained from light-field cameras is reasonably good enough to segment fence occlusions from images. Therefore, we chose Kinect v1, v2 and Lytro sensors as the main depth acquisition devices in this work. In order to provide a comparison with existing de-fencing algorithms, we use the passive stereo algorithm in [20] to compute depth maps.

Our approach using computational cameras (Kinect or Lytro) is closely related to existing work on image de-fencing [1–8] wherein only RGB data is used for occlusion segmentation. We divide the whole framework of image de-fencing and depth map completion into three sub-problems. For the given RGB-D data, firstly accurate segmentation of occluded regions of the static/dynamic scene captured using Kinect or Lytro is needed. Although several algorithms for this task have been proposed in literature [1–8], robust segmentation of occluded pixels from a single image is still a challenging problem. Note that overall performance of the image de-fencing and depth map completion framework depends on the robustness of the occluded pixel segmentation. In contrast to the works of [1–8], we exploit depth data to segment foreground occluded regions captured using computational cameras from the background scene. Note that the accuracy of occlusion segmentation depends on the robustness of the estimated depth. Unfortunately, no existing consumer-grade sensor is accurate [11]. Sometimes depth information obtained from Lytro or stereo algorithms [20] contains erroneous depth values at texture-less regions, which leads to wrong foreground occlusion masks. There are different feasible solutions for this issue: (a) combine depth from various sources to exploit complementary advantages,

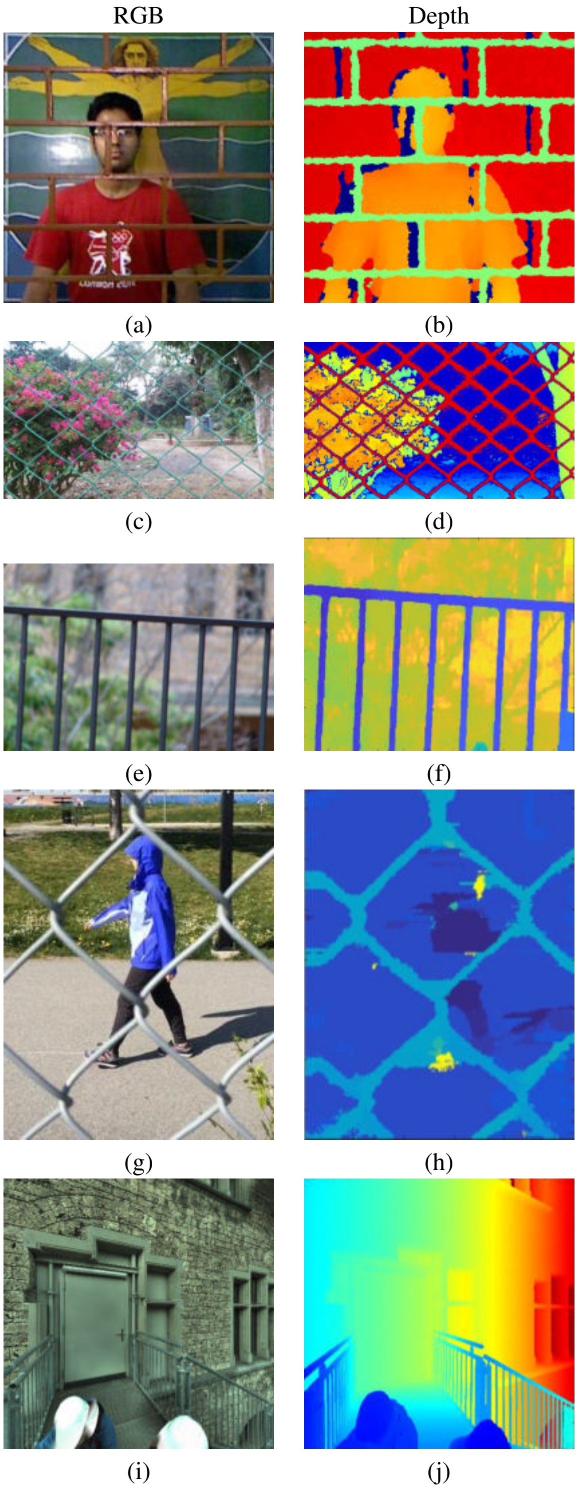


Fig. 2. Comparison of various depth acquisition techniques. From **a** to **j**: RGB images and corresponding depth maps estimated by (a, b) structured light (Kinect v1) [21], (c, d) Time-of-Flight (Kinect v2), (e, f) Lytro, (g, h) passive stereo algorithm [20], and (i, j) laser scanner, respectively. As can be seen in (b), shadows are observed in depth map obtained using Kinect v1 due to occlusions. Depth map obtained from Kinect v2 sensor is robust to occlusions, as is shown in (d). In (f), depth map obtained using light field data from the Stanford light field occlusion dataset [22] is shown. Disparity map obtained from passive stereo algorithm [20] shown in (h) is erroneous around depth edges. In (j), we show the depth map captured using laser scanner, which is taken from the dataset in [23]. It is robust to occlusions and is of high resolution.

(b) improve the individual depth acquisition technique, and (c) combine depth with other information channels. In this work for robust occlusion segmentation, we prefer to combine depth with RGB information to eliminate wrong segmentation due to erroneous depth values. To this end, we propose a deep end-to-end convolutional neural network (CNN) architecture for eliminating non-occluded regions.

Secondly, in contrast to image inpainting algorithms [9, 24], our proposed framework exploits the temporal cue for image and depth map completion. We observe that the information occluded in one frame will probably be available in neighboring frames due to the relative motion between the sensor (Kinect or Lytro) and the background scene. To exploit this temporal information for occlusion removal, it is necessary to compute the correspondences between the reference image and additional frames. Existing optical flow algorithms [25, 26] compute motion between *visible* pixels. However, fenced observations contain pixels belonging to the background scene which is occluded. If we use the estimated optical flow directly computed from the occluded observations using existing methods [25, 26], the proposed framework gives partially defenced images and incomplete depth maps. Therefore, there is a need for estimating motion between the occluded pixel and its corresponding visible pixel in the additional frame. Towards this end, in this work we propose an occlusion-aware optical flow algorithm under known occlusions for estimating the motion between reference and additional frames.

Finally, we fuse all the occluded observations to obtain a fence-free image and the completed depth map. Since the task of image de-fencing and depth map completion is an ill-posed inverse problem, we use total variation (TV) [27] of the de-fenced and completed depth maps as regularization constraints in the optimization framework. In order to preserve depth discontinuities along the boundaries of the depth layers, we also employ non-local regularization term on depth map along with the TV and solve the complete model using split Bregman iterative framework [28].

A. *Contributions*

The contributions of this paper can be summarized as follows:

- Existing works [1–8] address the issues of image defencing and depth map completion independently. To the best of our knowledge, we are the first to address both the problems in a single framework, in this paper.
- We harness depth/disparity cue for robust fence segmentation.
- We formulate an occlusion-aware optical flow algorithm under known fence occlusions to aid both de-fencing and depth map completion.
- We provide a joint optimization framework for simultaneous image and depth map de-fencing using temporal information from an RGB-D video sequence.
- We collect a light-field occlusion dataset, and provide the raw, depth maps obtained from a command line tool provided by Lytro.

We would like to point out that this paper is a comprehensive extension of our previously published works on image de-fencing [29, 30]. We point out that the proposed work possesses the following advantages: (1) In [29], we used only RGB information for the segmentation of fence occlusions based on deep learning, whereas in this work we harness depth cue for robust occlusion segmentation. (2) While the work in [30] used only stereo disparity maps, here we also exploit depth information estimated from various active sensing devices such as Kinect and Lytro cameras. (3) The work in [30] used an off-the-shelf recently-proposed motion estimation technique of [25]. However, we have now formulated an occlusion-aware optical flow algorithm which yields more accurate estimates of relative motion than [25] for fenced observations. (4) Moreover, the optimization methodology formulated in [29, 30] is considerably different from the one we follow here. We propose an objective function using the total variation of the de-fenced image and the completed depth profile as regularization constraints, which is optimized utilizing the split Bregman iterative framework. To preserve depth discontinuities along the boundaries of depth layers, we also integrate a non-local regularization term along with the local total variational prior on the depth map. (5) In contrast to [29, 30], our proposed methodology recovers occluded depth observations in addition to image de-fencing. (6) Finally, this paper contains extensive experimentation with additional datasets, as well as comparisons with recent approaches [2, 4] for both image de-fencing and depth completion.

This paper is organized as follows. We review related works in the literature in Section II. In Section III, we give details of proposed methodology including fence detection, motion computation under known fence occlusions and the optimization framework for information fusion. Experimental results and comparisons with the state-of-the-art algorithms are presented in Section IV. Finally, conclusions are given in section V.

II. Related Work

We observe that the problem we address is related to image/depth inpainting and image de-fencing, for which significant works have been reported in literature. We place our work in the context of these previous endeavours.

A. Image Inpainting

We refer readers to [31] for a comprehensive overview of image inpainting techniques. Here we limit our discussion to some of the state-of-the-art inpainting algorithms only. Conventional diffusion-based image inpainting techniques [24] propagate local information into the target region based on an isophote direction field. These algorithms work satisfactorily if the region with missing data is small in size and low-textured in nature. On the contrary, patch-based techniques [9, 32, 33] fill the occluded regions by using similar patches from other locations in the image. The advantage of methods belonging to this category is that they can recreate texture missing in large regions of the image.

The work of Kedar *et al.* [34] addressed video inpainting under constrained camera motion. The method in [35], used blurred images captured with three different aperture settings to remove thin occluders from images. The authors of [36] proposed a variational framework for image inpainting, and solved the problem using split Bregman technique. Kaimeng *et al.* [37] added another novel aspect to patch-based inpainting techniques, wherein statistics of patch offsets have been exploited. Recently, [38] proposed a context-aware inpainting algorithm by using the normalized histogram of Gabor responses as the contextual descriptors. Ebdelli *et al.* [39] proposed a video inpainting algorithm by grouping a small number of neighbor frames, and aligned them to inpaint a reference frame. Deep learning has also been applied to image inpainting tasks, and has achieved significant results [40–42]. Oord *et al.* [42] modeled discrete probabilities of raw pixels and encode dependencies in the image via recurrent neural networks. Very recently, Iizuka et al. [43], proposed an algorithm for large missing region completion in an image based on generative adversarial networks (GAN).

B. Image De-fencing

In recent years the problem of image de-fencing has attracted the attention of a significant number of researchers [1–5, 7]. Park et al. [44] detect fences from a single image by extracting near-regular repetitive patterns, under deformations due to perspective camera projection. The algorithm in [45] detected fence pixels using images with and without flash. Khasare *et al.,* [7] use interactive matting algorithm [46] for fence segmentation.

Mu *et al.* [2] harnessed parallax cue for the detection and removal of fences from video sequences. The work in [3] utilized motion cues for foreground segmentation, and restored the background using an optimization procedure. However, the algorithms in [2, 3] were limited to images containing static scene elements only. The works in [5, 8] proposed machine learning algorithms for fence texel joint detection, and formulated optimization frameworks to obtain the de-fenced image. Although these algorithms [5, 8] work for static and dynamic scenes, the fence detection algorithm therein work well only for standard fence patterns such as rhombic/square shapes.

Very recently, the work in [4] proposed an automatic fence segmentation algorithm using a video sequence. Initially, the method in [4] formulated a bottom-up approach for clustering pixels into coherent groups using color and motion features. Spatial structural features were exploited in [4] via graph-cut optimization to generate initial segmentation of fences. Finally, dense conditional random fields (CRF) employing multiple frames were used for improving segmentation results. For image de-fencing the algorithm in [4] employed exemplar-based image inpainting technique [9]. Although the work of [4] used multiple frames for fence segmentation, it does not exploit spatio-temporal information in the captured video for image de-fencing.

Apart from image-based techniques for image de-fencing, the method in [47] used RGB-D data for fence segmentation. The algorithm in [47] used depth data to generate foreground and background strokes which were fed to a graph-cut algorithm for accurate fence segmentation. However, the method

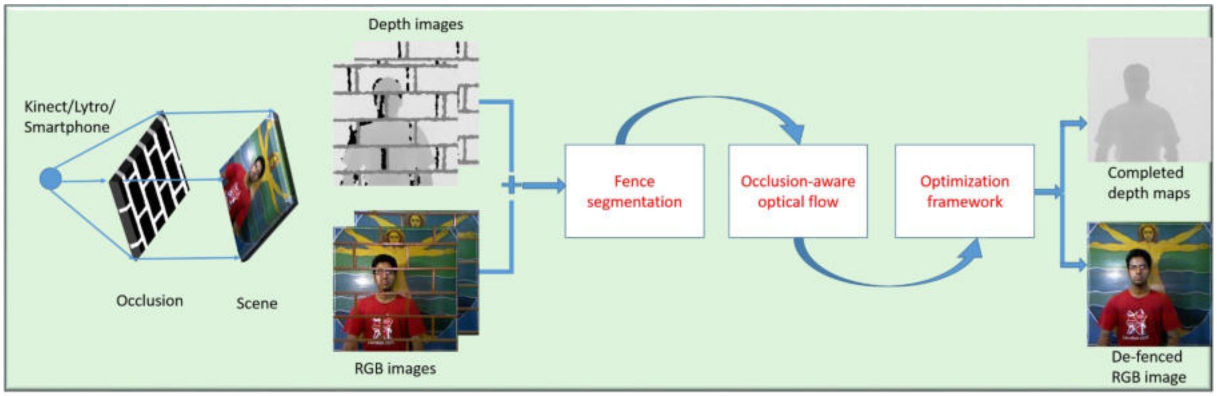


Fig. 3. Overall pipeline of the proposed image de-fencing and depth map completion algorithm.

in [47] resorted to image inpainting technique [9] for filling-in holes in RGB images.

In contrast to [4, 47], we exploit additional information in the frames of the captured video by proposing an occlusion-aware optical flow algorithm. Harnessing image data missing in the reference frame by relating adjacent frames with a degradation model, we obtain the de-fenced image by employing a robust optimization algorithm.

C. Depth Completion

Since we also perform depth completion in our work, we place it in the context of various depth inpainting algorithms in literature [48–52]. Liu et al. [49] proposed an extended fast marching algorithm for depth hole filling by using the aligned color image. Since the depth map is less textural than the corresponding color image, propagating depth information from the exterior to the interior of the inpainting region generally produces better results than diffusion does on color images. However, the difficulty lies in determining where to stop the propagation of depth values. The authors in [50] designed a weighting function to improve inpainting by considering geometrical distance, depth similarity, and structure information provided by the color image. Wang *et al.* [48] proposed an algorithm for the simultaneous completion of a color image and the corresponding depth map, which is closely related to our work. Their algorithm takes stereo images, estimates disparity maps, and fills in missing color and depth information due to occlusions. Finally, they used visible information from additional views and a modified version of the inpainting algorithm in [9] for simultaneous image and depth completion. In contrast to [48], our proposed algorithm exploits depth from various sources which allows users to capture RGB-D video sequences by a free-hand motion of the camera. Since the algorithm in [48] uses 3D warping for left and right image registration, it is limited to images containing static scene elements only. In this work, we propose an occlusion-aware optical flow algorithm for background image registration under known occlusions. Therefore, our framework can be applied to static/dynamic scene elements.

Lu *et al.* [51] presented an approach for simultaneous depth denoising and missing pixel completion, based on the observation that similar RGB-D patches lie in a low-dimensional subspace. They assembled similar patches into patch matrix and formulated depth map completion as an incomplete matrix factorization problem.

Recently, Park *et al.* [53] formulated a constrained optimization framework for depth map up-sampling and completion using a non-local structure prior for regularization. This fusion-based approach fills large holes in depth maps by considering structures and discontinuities from the corresponding registered RGB image. Buyssens *et al.* [54] proposed a superpixel-based algorithm for inpainting holes in depth maps which occur during RGB-D view synthesis. Dong *et al.* [55], presented a unified framework for accurate depth recovery by exploiting local and non-local color-depth dependencies. The authors of [52] proposed a low-rank low gradient regularization-based approach for depth map inpainting.

III. Methodology

A. Problem Formulation

We model the occluded RGB-D data as follows,

$$\mathbf{O}_p^x \mathbf{y}_p = \mathbf{y}_p^{\text{obs}} = \mathbf{O}_p^x[\mathbf{W}_p \mathbf{x} + \mathbf{n}_p^x] \qquad (1)$$

$$\mathbf{O}_p^d \mathbf{e}_p = \mathbf{e}_p^{\text{obs}} = \mathbf{O}_p^d[\mathbf{W}_p \mathbf{d} + \mathbf{n}_p^d] \qquad (2)$$

where the operators $\mathbf{O}_p^x$ and $\mathbf{O}_p^d$ crop out the un-occluded pixels from the p^{th} RGB and depth frames, $\mathbf{y}_p$ and $\mathbf{e}_p$ represent the occluded RGB and depth observations used, respectively, $\mathbf{W}_p$ is the warp matrix, $\mathbf{x}$ and $\mathbf{d}$ are the completed image and depth map, respectively. And $\mathbf{n}_p^x$ and $\mathbf{n}_p^d$ are the noise assumed as Gaussian. As we discussed in Section 1, the problem of image de-fencing/depth map completion was divided into three sub-problems and the overall workflow of the proposed algorithm is shown in Fig. 3.

B. Depth Cue for Occlusion Segmentation

In contrast to image inpainting techniques [9, 24, 41] which assume that the region to be inpainted is specified a-priori. The crucial task in the proposed framework is to segment occluded pixels corresponding to the operators $\mathbf{O}_p^x$ and $\mathbf{O}_p^d$ in Eqs. 1 and 2, respectively. Although, several fence detection algorithms have been proposed in literature that use RGB data [1–5], separating occlusions robustly from a single image is still a challenging problem. Since obstructions in real-world scenarios exhibit diverse shapes, color variations, illumination changes, view points, etc., it is very difficult to accurately segment occlusions from images of static/dynamic scenes elements using RGB data only. Interestingly, we observe that if one captures any scene by panning a camera, occlusions are almost always closer to the sensor. We harness *depth/disparity* cue for robust fence/occlusion segmentation from a single color image and the corresponding depth map. Presently, depth or 3D information along with RGB image is acquired using low-cost computational cameras such as Kinect v1, Kinect v2 and Lytro as mentioned in Section 1. The same is possible with next-generation *smartphones* [56] which will soon be accessible to common masses. In order to report comparison results with state-of-the-art fence removal algorithms [2, 4], we also estimate the disparity map from a pair of images.

We capture RGB-D video sequences using both structured-light (Kinect v1) and Time-of-Flight (Kinect v2) depth cameras. Here, in this work, with the availability of depth information in addition to RGB image, we can segment foreground obstructions automatically. Since in Kinect both the RGB and IR cameras have different fields of view due to an offset between color and depth camera, it is important to register color and depth frames before using depth maps for obstruction segmentation. To obtain the aligned depth map, we have used ‘CoordinateMapper’ function from Microsoft’s software development kit (SDK). In Figs. 4 (a), (b) we show the color and depth images captured using Kinect v2 sensor. The depth map aligned with respect to the color image in Fig. 4 (a) is shown in Fig. 4 (c).

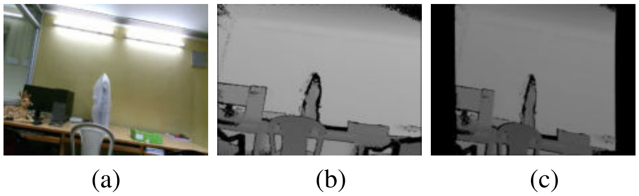


Fig. 4. (a), (b) RGB image and its corresponding depth map captured using Kinect v2 (TOF) sensor. (c) Depth map in (b) aligned with respect to the color image in (a).

Due to the requirement of huge bandwidth for light field data capture, it is infeasible to record video sequences with the existing consumer light field cameras [57]. In our work to simulate light field video data, we captured 4D light field at different spatial locations by panning the Lytro camera relative to the 3D scene. In Figs. 5 (a), (b) we show the processed sub-aperture image and its corresponding depth profile obtained from the software tool provided by Lytro. Depth information allows us to easily separate foreground occlusions from the background non-occluded pixels. The simplest approach is to segment all the pixels below a threshold value T_{fg} as initial occluded pixels and others as background pixels. We choose a threshold value T_{fg} from the depth histogram in Fig. 5 (c). In Fig. 5 (d), we show the initial occlusion mask obtained from Fig. 5 (b) using T_{fg}.

$$O_{init}(i) = \begin{cases} 1, & \text{if } e(i) \leq T_{fg} \\ 0, & \text{Otherwise} \end{cases} \tag{3}$$

where $e(i)$ is the depth value corresponding to pixel i. Most of the existing sources for estimating depth (Kinect/Lytro or stereo) produce erroneous results around the boundaries, and in texture-less regions.

As discussed in section 1, Kinect data has imperfections near depth discontinuities, light-field data contains erroneous regions at texture-less segments, and stereo disparity suffers from occlusions and texture-less regions. To support our claim, due to erroneous depth values in Fig. 5 (b), the non-occluded pixels are also classified as occluded pixels depicted with a red colored window in Fig. 5 (d). Since accurate depth plays a major role in occlusion segmentation (followed by removal), we need to eliminate non-occluded pixels from Fig. 5 (d). There are three possible solutions, (1) improve the depth estimation techniques, (2) use depth fusion to obtain enhanced depth, since different depth sensors have complementary advantages, and (3) combine depth with other information channels for occlusion refinement. In our work, we choose to combine other information (RGB) channels for occlusion refinement. However, RGB and depth channels carry conflict information [58], combining them directly is very hard. In this work, we prefer to combine RGB data with depth information for occlusion refinement using end-to-end convolution neural network (CNN) architecture.

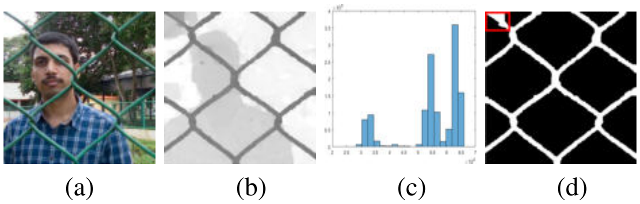


Fig. 5. (a), (b) RGB image and its corresponding depth map captured using Lytro. (c) Histogram corresponding to depth map in (b). (d) Binary fence mask obtained by depth map thresholding.

C. Semantic Segmentation and Occlusion Refinement

To eliminate false segmentation regions due to erroneous depth obtained from light-field or stereo disparity, we use segmentation mask predicted from the end-to-end CNN architecture. The overview of the segmentation network architecture is shown in Fig. 6. It consists of 6 layers, three convolution (i.e. layers 1,4, and 5), two dilated-convolution (layers 3 and 4) and final layer being the prediction layer. Dilated convolution layers [59] operates at larger spatial views to generate more accurate predictions while maintaining the same computational complexity. Dilation factors for both the dilated convolution layers are 2, 4, respectively. All the layers in the network shown in Fig. 6 followed by a ReLU layer. Each layer consists of 32 convolution kernels of size 3 × 3 followed by a ReLU layer. Finally, the last prediction layer predicts the segmentation map.

Suppose we are given a supervised occlusion segmentation task and a training set of input-target pairs $\mathbf{x}^{(i)}, \mathbf{y}^{(i)}$. Our

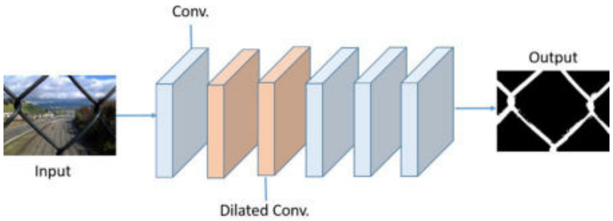


Fig. 6. Overview of the end-to-end CNN architecture for learning occlusion segmentation task.

objective is to learn the parameters θ of a representation function G_θ which optimally approximates the input-target dependency according to a loss function $\mathscr{L}(G_\theta(\mathbf{x}), \mathbf{y})$. Typical choices are mean squared error (MSE) loss

$$\mathscr{L}(G_\theta(\mathbf{x}), \mathbf{y}) = \parallel G_\theta(\mathbf{x}) - \mathbf{y} \parallel_2^2 \tag{4}$$

or ℓ_1 loss

$$loss = \parallel G_\theta(\mathbf{x}) - \mathbf{y} \parallel \tag{5}$$

In order preserve peace wise smoothness in the predictions, we also added total variation regularization term in addition to MSE loss in Eq. (4). The combined loss function used in the end-to-end network is as given as

$$\mathscr{L}(G_\theta(\mathbf{x}), \mathbf{y}) = \parallel G_\theta(\mathbf{x}) - \mathbf{y} \parallel_2^2 + \lambda_{tv} \parallel \nabla G_\theta(\mathbf{x}) \parallel_1 \tag{6}$$

where λ_{tv} is the regularization parameter, whose value is tuned as $\lambda_{tv} = 1e - 3$ by experiments. Initially, we have a base dataset of 250 RGB images and its corresponding ground truth occlusion masks generated using the algorithm in [46] with significant user intervention. Since it is required to have large datasets for training CNNs, we have augmented the base dataset and created an augmented dataset of 500 images using flipping. To train this end-to-end CNN, we divide the set of 500 images into sub-images of size 128 × 128 and its corresponding ground truth segmentation masks. Before training the network, weights are initialized with Xavier initialization. The learning rate is changed over epochs, starting with a higher learning rate $2e^{-3}$ and decreased during training to $1e^{-6}$. We trained the network for 500 epochs. During the testing phase, we feed the input image of any arbitrary size through the trained net and obtain the segmentation map O_{seg} of the same resolution.

In Fig. 7 (a), we show the image taken from the video sequence of [2]. The stereo disparity between Fig. 7 (a) and its neighbor frame is shown in Fig. 7 (b). Due to texture-less regions and occlusions stereo disparity map is unreliable at some locations marked with red colored windows in Fig. 7 (b). The segmentation map obtained using proposed end-to-end CNN is shown in Fig. 7 (c), which is reasonable good at all the locations except few regions marked with red colored boxes in Fig. 7 (c). It is observed that, erroneous regions in Fig. 7 (b) and (c) are complimentary. Finally, we obtain the refined occlusion mask, by combining the complimentary advantages of both segmentation map from depth and end-to-end CNN. We apply logical AND operation between Fig. 7 (b) and (c), to obtain Fig. 7 (d) which corresponds to the operator O_p^x in Eq. (1).

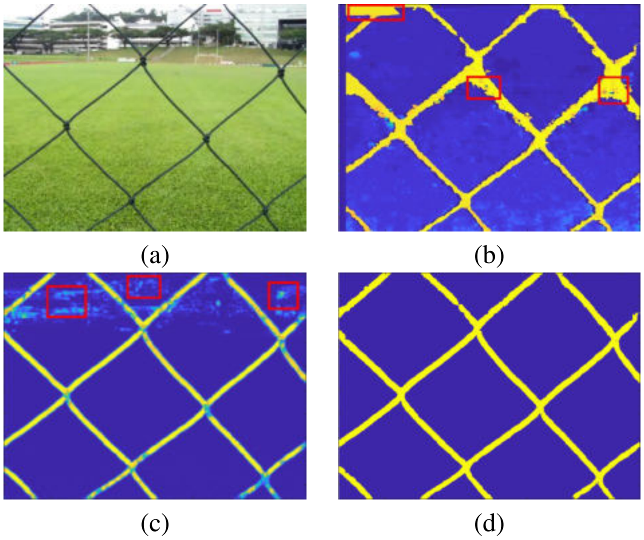


Fig. 7. (a) RGB image taken from the video sequence reported in [2]. (b) Stereo disparity computed between (a) and additional frame from the same video sequence. (c) Segmentation mask obtained from proposed end-to-end CNN. (d) Final binary occlusion mask corresponding to (a) obtained by logical AND operation between (b) and (c).

As we discussed in section 1, depth maps captured using Kinect v1 and v2 contains noisy or missing pixels. In Fig. 8 (a), (b), we show the color image and its corresponding depth map captured using Kinect v1 sensor. Missing depth information depicted in black color in Fig. 8 (b) is due to the offset between IR projector and camera. The segmentation mask corresponding to external occlusion obtained using Eq. (3) is shown in Fig. 8 (c). And in Fig. 8 (d), we present the binary mask corresponding to missing or noisy pixels in Fig. 8 (b). The final binary mask corresponding to O_p^d operator in Eq. (2) is shown in Fig. 8 (e) obtained by applying logical OR operation between Figs. 8 (c) and (d).

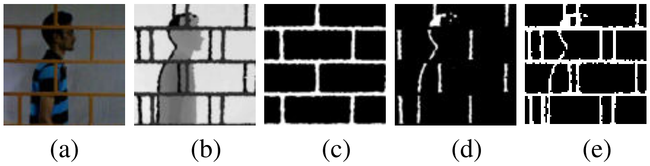


Fig. 8. (a), (b) Color image and its corresponding occluded depth map captured using Kinect v1 sensor. (c) Binary occlusion mask obtained using Eq. (3) corresponding to external occlusions. (d) Binary mask corresponding to missing pixels in (b). (e) Final binary mask obtained by combining both (c) and (d).

In Fig. 9 (a), (b), we present the processed central view of the color image and its corresponding depth map obtained from light field data captured using Lytro camera. The binary segmentation mask generated from Fig. 9 (b) using Eq. (3) is shown in Fig. 9 (c). Next, we feed input occluded color observation to the trained end-to-end CNN in Fig. 6 and the predicted segmentation map depicted in Fig. 9 (d). Finally, we obtain the refined binary occlusion mask corresponding to the operator O_p^x in Eq. (1) by combining the complimentary advantages of both the approaches.

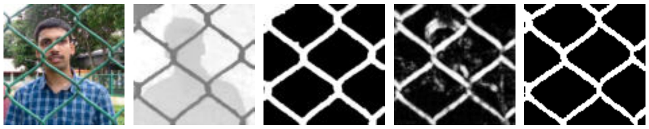


Fig. 9. (a), (b) RGB image and its corresponding depth profile obtained from the light field data captured using Lytro. (c) Binary segmentation mask obtained from (b) using Eq. (3). (d) Segmentation mask generated by feeding (a) to the proposed end-to-end CNN. (e) Final binary occlusion mask obtained by combining both (c) and (d), respectively.

D. Motion Estimation under Known Fence Occlusions

In this work, we use multiple frames from a video sequence captured by the free-hand motion of the camera. As mentioned in section 1, information occluded in the reference image will most probably be revealed in neighboring frames of the video sequence. For filling-in the occluded pixel in the reference image, we need to compute the correspondence of all pixels between two images despite occlusions. Although, many optical flow techniques [25, 60, 61] have been proposed in the literature, they cannot be directly used in the problem considered here. The methods in [25, 60, 61] compute the motion between visible pixels among the frames only.

In this paper, we re-formulate the optical flow algorithm of [26] to fit our application of image and depth map completion. Akin to [26], coarse to fine optical flow is estimated using an incremental framework in Gaussian scale-space. Note that we have already estimated the binary occlusion mask $\mathbf{O}_p^x$ corresponding to the occluded pixels in the observation $\mathbf{y}_p$. We insert this mask $\mathbf{O}_p^x$ as occlusion operator inside the optical flow framework to deal with the motion inaccuracies at occluded pixel locations. At the occlusion locations, data cost is assumed to be zero and only smoothness term guides optical flow estimation. We assume total variation prior for both horizontal and vertical velocities. At every scale, the estimated optical flow values are up-scaled and used as an 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 [26, 62], one needs to estimate the best increment $d\mathbf{w} = (du, dv)$ as follows

$$\begin{aligned} E(du, dv) = \arg\min_{d\mathbf{w}} \parallel \mathbf{F}_{\mathbf{w}+d\mathbf{w}}\tilde{y}_t - \tilde{y}_r \parallel_1 \\ +\mu \parallel \nabla(u+du) \parallel_1 +\mu \parallel \nabla(v+dv) \parallel_1 \end{aligned} \tag{7}$$

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 $\mathbf{O} = \mathbf{F}_{\mathbf{w}+d\mathbf{w}}\mathbf{O}_t^x || \mathbf{O}_r^x$ obtained by the logical OR operation between the fence mask from the reference image $\tilde{y}_r$ 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. (4) as

$$\begin{aligned} E(du, dv) = \arg\min_{d\mathbf{w}} \parallel \mathbf{O}(\mathbf{F}_{\mathbf{w}+d\mathbf{w}}\tilde{y}_t - \tilde{y}_r) \parallel_1 \\ +\mu \parallel \nabla(u+du) \parallel_1 +\mu \parallel \nabla(v+dv) \parallel_1 \end{aligned} \tag{8}$$

By first-order Taylor series expansion,

$$\mathbf{F}_{\mathbf{w}+d\mathbf{w}}\tilde{y}_t \approx \mathbf{F}_{\mathbf{w}}\tilde{y}_t + \mathbf{Y}_x du + \mathbf{Y}_y dv \tag{9}$$

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. (5) as

$$\begin{aligned} \arg\min_{d\mathbf{w}} \parallel \mathbf{O}\mathbf{F}_{\mathbf{w}}\tilde{y}_t + \mathbf{O}\mathbf{Y}_x du + \mathbf{O}\mathbf{Y}_y dv - \mathbf{O}\tilde{y}_r) \parallel_1 \\ +\mu \parallel \nabla(u+du) \parallel_1 +\mu \parallel \nabla(v+dv) \parallel_1 \end{aligned} \tag{10}$$

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.

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

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 $\mathbf{D}_y$ as 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.

E. Optimization Framework for RGB-D Completion

To harness the complimentary nature of information in RGB-D data, we formulate an integrated framework for image and depth map completion. Due to ill-posedness of the problem, we employ total variation (TV) of the de-fenced image and inpainted depth maps as the regularization constraints in order to preserve geometry in addition to occlusion removal. Total variation regularization is a well-studied approach which preserves discontinuities in the reconstructed image [36]. We tune regularization parameters μ_x, μ_d, and μ_{nl} to obtain the best estimate of the de-fenced image and inpainted depth map.

Given the discrete image $\mathbf{x} \in \mathbb{R}^{M \times N}$, there two different choices for total variation in the literature, namely, isotropic TV which is defined by

$$TV_{iso} = \sum_{i=1}^{M}\sum_{j=1}^{N}\sqrt{(x_{i,j} - x_{i+1,j})^2 + (x_{i,j} - x_{i,j+1})^2} \tag{11}$$

and the anisotropic TV defined by

$$TV_{aniso} = \sum_{i=1}^{M}\sum_{j=1}^{N}\{\left|x_{i,j} - x_{i+1,j}\right| + \left|x_{i,j} - x_{i,j+1}\right|\} \tag{12}$$

The depth data captured using Kinect v1 and v2 sensors generally contains holes and noisy pixels. Moreover, the physical offset between color and depth cameras leads to holes in the depth maps at boundaries of objects in the scene. We complete the holes due to Kinect sensor or external occlusions with the aid of temporal information since we have captured RGB-D video data. However, depth layer boundaries in the de-fenced depth map still suffer from inconsistencies. Since depth maps contain enough redundancy, it motivates us to incorporate non-local depth regularization prior in addition to the local smoothness term in the proposed objective function. It is possible to find more non-local self-similarities within a larger neighborhood in the depth map, utilizing which the quality of depth reconstruction near the boundaries can be enhanced by minimizing the artifacts which appear due to usage of only local smoothness prior. We use RGB-D information for computing the similarity between the reference and non-local patch [63]. Given a depth map $\mathbf{d} \in \mathbb{R}^{M \times N}$ the non-local depth regularization term is defined as

$$E_{NL}(\mathbf{d}) = \sum_{i=1}^{MN} \sum_{j \in \mathcal{N}(i)} \frac{s_{ij}}{S_i} (\mathbf{d}(i) - \mathbf{d}(j))^2 \tag{13}$$

where $\mathcal{N}(i)$ is a local window in the restored depth map, s_{ij} represents the similarity score between neighboring pixels i, j and $S_i = \sum_j s_{ij}$ represents the normalization score. The weighting term s_{ij} is computed based on two scores: color similarity and depth similarity. The authors in [63] mentioned that due to the co-occurrence of edges in RGB-D data, the color term can help avoid depth discontinuities whereas depth term to prevent from recovering incorrect depth due to depth-color inconsistency. The depth and color terms are defined using Gaussian function as follows

$$s_{ij}^{d} = \exp(-\frac{(\mathbf{d}_i - \mathbf{d}_j)^2}{2\sigma_d^2}), s_{ij}^{x} = \exp(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{2\sigma_x^2}) \tag{14}$$

where $\mathbf{d}_i, \mathbf{x}_i$ representing depth and intensity vectors obtained from a patch of size 7×7 around a pixel i, σ_d, σ_x are standard deviations. Therefore, the similarity score is computed as a product of both depth and color terms $s_{ij} = s_{ij}^{d} s_{ij}^{x}$. The defenced image and completed depth map are the solutions of the following optimization framework

$$\arg\min_{\mathbf{x},\mathbf{d}} = E_d(\mathbf{x}) + E_d(\mathbf{d}) + E_{TV}(\mathbf{x}) + E_{TV}(\mathbf{d}) + E_{NL}(\mathbf{d}) \tag{15}$$

where $\mathbf{x}$, $\mathbf{d}$ are de-fenced image and completed depth profiles. $E_d(\mathbf{x})$, $E_d(\mathbf{d})$ and $E_{TV}(\mathbf{x})$, $E_{TV}(\mathbf{d})$ are data fidelity and regularization terms corresponding to RGB and depth maps, respectively. And $E_{NL}(\mathbf{d})$ represents the color-guided non-local regularization term for thin structure recovery in depth maps.

$$\begin{aligned} \arg\min_{\mathbf{x},\mathbf{d}} \frac{1}{2} \sum_{p=1}^{P} [\parallel \mathbf{y}_p - \mathbf{O}_p^x \mathbf{W}_p \mathbf{x} \parallel_2^2 + \parallel \mathbf{e}_p - \mathbf{O}_p^d \mathbf{W}_p \mathbf{d} \parallel_2^2] \\ + \mu_x \parallel \nabla \mathbf{x} \parallel_1 + \mu_d \parallel \nabla \mathbf{d} \parallel_1 + \mu_{nl} \parallel \mathbf{Sd} \parallel_2^2 \end{aligned} \tag{16}$$

where P is the number of frames chosen from the video and μ_x, μ_d, μ_{nl} are the regularization parameters. The above problem can also be written in a constrained framework as

$$\begin{aligned} \arg\min_{\mathbf{x},\mathbf{d}} \frac{1}{2} \sum_{p=1}^{P} [\parallel \mathbf{y}_p - \mathbf{O}_p^x \mathbf{W}_p \mathbf{x} \parallel_2 + \parallel \mathbf{e}_p - \mathbf{O}_p^d \mathbf{W}_p \mathbf{d} \parallel_2] \\ + \mu_x \parallel \mathbf{a} \parallel_1 + \mu_d \parallel \mathbf{b} \parallel_1 + \mu_{nl} \parallel \mathbf{Sd} \parallel_2^2 \; s.t. \;\; \mathbf{a} = \nabla \mathbf{x}, \;\; \mathbf{b} = \nabla \mathbf{d} \end{aligned} \tag{17}$$

To enforce the constraints, we add the penality terms in the equation above as follows:

$$\begin{aligned} \arg\min_{\mathbf{x},\mathbf{d}} \frac{1}{2} \sum_{p=1}^{P} [\parallel \mathbf{y}_p - \mathbf{O}_p^x \mathbf{W}_p \mathbf{x} \parallel_2 + \parallel \mathbf{e}_p - \mathbf{O}_p^d \mathbf{W}_p \mathbf{d} \parallel_2] \\ + \mu_x \parallel \mathbf{a} \parallel_1 + \mu_d \parallel \mathbf{b} \parallel_1 + \lambda_x \parallel \mathbf{a} - \nabla \mathbf{x} \parallel \\ + \lambda_d \parallel \mathbf{b} - \nabla \mathbf{d} \parallel + \mu_{nl} \parallel \mathbf{Sd} \parallel_2^2 \end{aligned} \tag{18}$$

The Bregman iterates to solve the above are as follows:

$$\begin{aligned} [\mathbf{x}^{k+1}, \mathbf{d}^{k+1}, \mathbf{a}^{k+1}, \mathbf{b}^{k+1}] = \arg\min_{\mathbf{x},\mathbf{d},\mathbf{a},\mathbf{b}} \; \frac{1}{2} \sum_{p=1}^{P} [\parallel \mathbf{y}_p - \mathbf{O}_p^x \mathbf{W}_p \mathbf{x} \parallel_2^2 \\ + \parallel \mathbf{e}_p - \mathbf{O}_p^d \mathbf{W}_p \mathbf{d} \parallel_2^2] + \mu_x \parallel \mathbf{a} \parallel_1 + \mu_d \parallel \mathbf{b} \parallel_1 + \lambda_x \parallel \mathbf{a} - \nabla \mathbf{x} \\ + p^k \parallel_2^2 + \lambda_d \parallel \mathbf{b} - \nabla \mathbf{d} + q^k \parallel_2^2 + \mu_{nl} \parallel \mathbf{Sd} \parallel_2^2 \end{aligned} \tag{19}$$

$$\mathbf{p}^{k+1} = \mathbf{p}^k + \nabla \mathbf{x}^{k+1} - \mathbf{a}^{k+1} \tag{20}$$

$$\mathbf{q}^{k+1} = \mathbf{q}^k + \nabla \mathbf{d}^{k+1} - \mathbf{b}^{k+1} \tag{21}$$

We can now split the above problem into sub problems as follows:

Sub Problem 1:

$$[\mathbf{x}^{k+1}] = \arg\min_{\mathbf{x}} \; \frac{1}{2} \sum_{p=1}^{P} \parallel \mathbf{y}_p - \mathbf{O}_p^x \mathbf{W}_p \mathbf{x} \parallel_2^2 + \lambda_x \parallel \mathbf{a}^k - \nabla \mathbf{x} + \mathbf{p}^k \parallel_2^2 \tag{22}$$

Sub Problem 2:

$$\begin{aligned} [\mathbf{d}^{k+1}] = \arg\min_{\mathbf{d}} \; \frac{1}{2} \sum_{p=1}^{P} \parallel \mathbf{e}_p - \mathbf{O}_p^d \mathbf{W}_p \mathbf{d} \parallel_2^2 \\ + \lambda_d \parallel \mathbf{b}^k - \nabla \mathbf{d} + \mathbf{q}^k \parallel_2^2 + \mu_{nl} \parallel \mathbf{Sd} \parallel_2^2 \end{aligned} \tag{23}$$

The above sub problems 1 and 2 are solved by CG algorithm.

Sub Problem 3:

$$[\mathbf{a}^{k+1}] = \arg\min_{\mathbf{a}} \mu_x \parallel \mathbf{a} \parallel_1 + \lambda_x \parallel \mathbf{a} - \nabla \mathbf{x}^{k+1} + \mathbf{p}^k \parallel_2^2 \tag{24}$$

Sub Problem 4:

$$[\mathbf{b}^{k+1}] = \arg\min_{\mathbf{b}} \mu_d \parallel \mathbf{b} \parallel_1 + \lambda_d \parallel \mathbf{b} - \nabla \mathbf{d}^{k+1} + \mathbf{q}^k \parallel_2^2 \tag{25}$$

The above sub problems 3 and 4 can be solved by applying the shrinkage operator as follows:

$$\begin{aligned} \mathbf{a}^{k+1} &= shrink(\nabla \mathbf{x}^{k+1} + \mathbf{p}^k, \frac{\lambda_x}{\mu_x}) \\ \mathbf{b}^{k+1} &= shrink(\nabla \mathbf{d}^{k+1} + \mathbf{q}^k, \frac{\lambda_d}{\mu_d}) \\ where \; shrink(\mathbf{d}, \lambda) &= \frac{\mathbf{d}}{\mid \mathbf{d} \mid} * max(\mid \mathbf{d} \mid -\lambda, 0) \end{aligned} \tag{26}$$

the update for p and q are as follows,

$$\begin{aligned} \mathbf{p}^{k+1} &= \nabla \mathbf{x}^{k+1} + \mathbf{p}^k - \mathbf{a}^{k+1} \\ \mathbf{q}^{k+1} &= \nabla \mathbf{d}^{k+1} + \mathbf{q}^k - \mathbf{b}^{k+1} \end{aligned} \tag{27}$$

IV. Experimental Results

We evaluate the proposed algorithm on different RGB-D datasets obtained using various sources. In this work, we report results on both generations of the Kinect sensor (i.e., Kinect v1 and v2) and light-filed information captured using Lytro camera. Moreover, we captured a light-field occlusion data set with first generation Lytro camera and provided the raw information, depth maps obtained from a command line tool provided by Lytro. To produce fair comparisons with the state-of-the-art image de-fencing techniques [2, 4] and show the effectiveness of our optimization framework which uses temporal information for dis-occlusion, we obtain fence masks corresponding to RGB video sequence reported in [2, 4] using a stereo algorithm [20]. We also provided the comparison with the existing image inpainting techniques [9, 64]. We ran all our experiments on a 3.6GHz Intel Core i7 processor with 16 GB of RAM and 4 GB Nvidia GeForce GTX 745.

Firstly, we report results on the RGB-D video sequence captured using Kinect v1 sensor. In Fig. 10 (a), (b) we show the RGB and corresponding depth images captured using Kinect v1 sensor. Fence mask corresponding to occluded observations are obtained using our algorithm. Under known fence pixels motion between the frames estimated using occlusion-aware optical flow algorithm. Subsequently, de-fenced image and completed depth maps are obtained by solving split Bregman iterative framework and shown in Figs. 10 (c), (d), respectively. We observe that hardly there are any artifacts in (c) and (d).

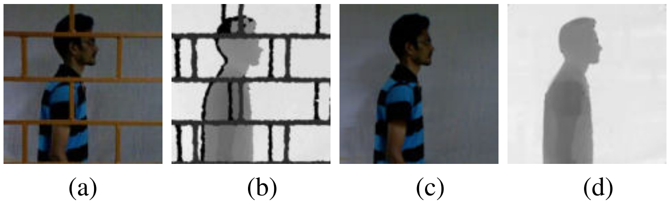


Fig. 10. **Kinect v1 dataset**: (a), (b) RGB and its corresponding occluded depth profile captured using Kinect v1 sensor. (c) De-fenced image using the proposed method. (d) Completed depth map using our method.

Here we evaluate proposed algorithm on the dataset captured using Kinect v2 sensor. In Figs. 11 (a), (b), we show the RGB and its corresponding depth images captured using Kinect v2 sensor. We segment the fence pixels using the proposed algorithm. The pixel correspondence between the frames computed using proposed occlusion-aware optical flow algorithm. De-fenced and inpainted depth map obtained using split Bregman iterative framework are shown in Figs. 11 (c), (d), respectively. Next, we captured a video sequence of a person walking behind a fence. In Figs. 12 (a), (b), we show the two frames taken from that sequence. The optical flow computed by [25] algorithm between the frames in (a), (b) is in Fig. 12 (c). Motion obtained using proposed occlusion-aware optical flow algorithm is shown in Fig. 12 (d). The de-fenced images corresponding to the motion in (c), (d) are shown in Figs. 12 (e), (f), respectively. We observe that there some pixels are failed to restore in (e). Note that occlusion-aware motion estimation algorithm allows us to use temporal information from the additional frames whereas LDOF [25] match between fence pixels in two frames. In Figs. 12 (g) and (h) we show the occluded depth map and completed depth profile obtained using the proposed algorithm, respectively.

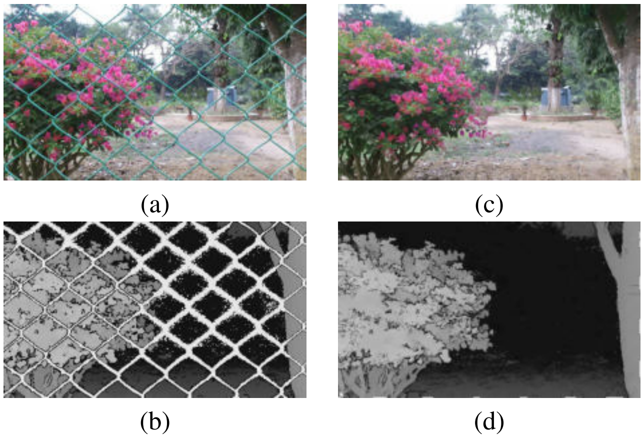


Fig. 11. **Kinect v2 dataset**: (a), (b) Color image and its corresponding aligned depth map captured using Kinect 2 sensor. (c), (d) De-fenced and inpainted depth images obtained using our algorithm corresponding to (a), (b), respectively.

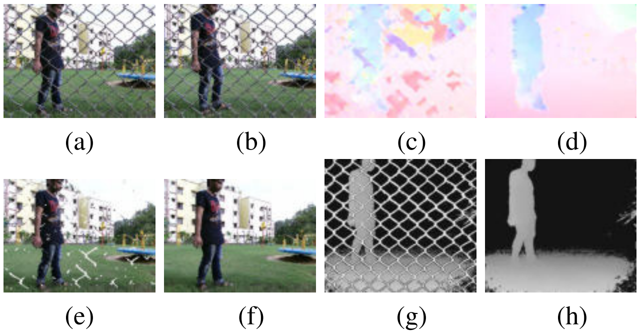


Fig. 12. **Kinect v2 dataset**: (a), (b) Two images taken from a video sequence captured using Kinect v2 sensor. (c), (d) Optical flow estimated between (a), (b) without and with occlusion handling, respectively. (e), (f) De-fenced images obtained using proposed algorithm with the optical flow in (c), (d), respectively. (g), (h) Occluded depth profile and its corresponding completed depth map obatined using proposed algorithm.

Light-field Obstruction Data: As we discussed in the previous section, depth information is useful in many computer vision and machine learning problems. Here we exploited the depth encoded in 4D light field information for robust foreground obstruction segmentation. In contrast to conventional cameras, a light-field camera has the capability of capturing angular information along with the intensity. We captured a light-field obstruction data set with first generation Lytro camera and provided the raw information, depth maps obtained from a command line tool provided by Lytro. In this work, we extract the processed sub-aperture images and corresponding depth maps encoded in light-field information using a command line tool provided by Lytro. In the first row of Fig. 13, we show the processed sub-aperture images captured using Lytro camera. The depth maps obtained from light-field information are shown in the second row of Fig. 13. We segment the foreground occlusions using the proposed algorithm and are depicted in third row of Fig. 13. We compute the motion between reference frame in first row and the additional frame from captured RGB-D sequence using formulated occlusion-

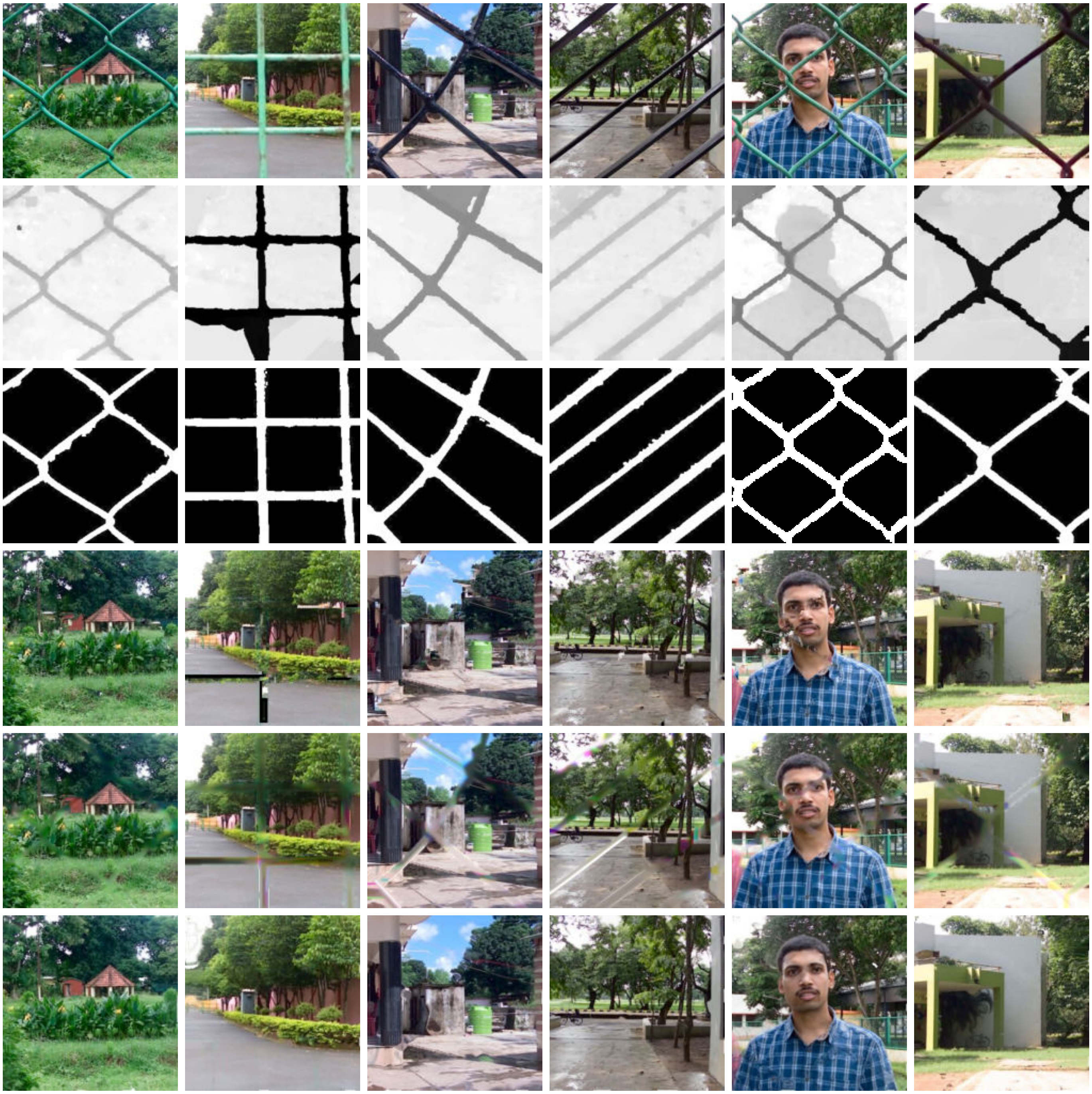


Fig. 13. **Results on light-field occlusion dataset**. First row: processed sub-aperture images from the light-filed information captured using Lytro camera. Second row: Depth maps obtained from a command line tool provided by Lytro. Third row: binary occlusion segmentation masks obtained using proposed algorithm corresponding to images in the first row. Fourth row: inpainting results obtained using exemplar-based inpainting technique [9]. Fifth row: inpainted results computed using the algorithm in [64]. Sixth row: de-fencing results obtained using proposed algorithm corresponding to images in the first row.

aware optical flow algorithm. In the fourth row of Fig. 13, we show the inpainted images obtained using exemplar-based region-filling technique in [9]. We observed that the images in the fourth row of Fig. 13 obtained using the exemplar-based inpainting algorithm, contain several artifacts. The inpainted results corresponding to first row of images obtained using [64] are shown in the fifth row of Fig. 13. Note that algorithm in [64] failed to recover the texture information behind occlusions. Finally, the obstruction-free images obtained using proposed algorithm are shown in last row of Fig. 13. As the proposed framework uses temporal information, we could avoid such artifacts, the same is evident in the de-fencing results shown in the last row of Fig. 13. For all our experiments we used three frames.

Stereo Disparity: In order to provide a fair comparison with the recent image de-fencing algorithm [2, 4] which uses RGB data only, we generate the disparity maps corresponding to RGB images using the technique proposed in [20]. We obtain the occluded pixels from disparity map with slight user interaction. In Figs. 14 (a), (b), we show the two of the images taken from the video sequences reported in [2, 4] used in our experiment. The stereo disparity maps generated using the algorithm [20] shown in Fig. 14 (c). In Fig. 14 (d), we present the color coded optical flow between (a), (b) obtained using the variational framework in [25]. The de-fenced image estimated using the flow in (d) is shown in Fig. 14 (e). The occlusion-aware optical flow obtained using the proposed algorithm is shown in Fig. 14 (f). We show the inpainted image obtained using [9] corresponding to (a) is shown in Fig. 14 (g). The de-fenced images obtained using proposed algorithm with occlusion-aware optical flow is shown in Fig. 14 (h). We perceive that there are some artifacts marked with yellow colored circles in Fig. 14 (g), whereas the proposed algorithm de-fence the occluded pixels completely in Fig. 14 (h) since it uses temporal information.

We provide a comparison with state-of-the-art image de-fencing algorithm which used depth cue for foreground segmentation [47]. According to our knowledge in addition to our preliminary work [65] there was only one algorithm in [47] used depth data for fence segmentation. In Fig. 15 (a), we show the RGB image taken from a video sequence reported in [47]. The de-fenced images obtained using the method in [47] and our algorithm is shown in Figs. 15 (b), (c), respectively. We observe that the algorithm in [47] failed to restore the image at some places whereas proposed algorithm completely de-fenced the image since we are using temporal information. The same can be depicted with yellow colored circles in Figs. 15 (b), (c), respectively.

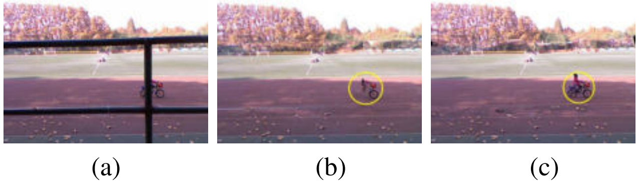


Fig. 15. Comparison with [47]: (a) An image taken from a video sequence reported in [47]. (b) De-fenced image corresponding to (a) using the algorithm in [47]. (c) De-fenced image obtained using proposed algorithm.

V. Conclusions

We proposed a multi-modal approach for an image de-fencing and depth map completion of a scene using RGB-D data captured using computational cameras. Specifically, we use multiple frames from a video and the aligned depth maps captured by panning the scene with the Kinect, Lytro and smartphone cameras. We addressed the crucial task of robust segmentation of the fence pixels by using the captured color and depth data. Next, we addressed the pixel correspondence estimation problem via proposed occlusion-aware optical flow algorithm. Finally, a joint framework for both image de-fencing and depth map completion has formulated using the total variation of estimates as regularization constraints. To preserve the depth discontinuities across boundaries we also integrated non-local depth regularization term in addition to local variational prior. To show the efficacy of the proposed methodology, we provide comparisons with the existing in-painting and state-of-the-art de-fencing techniques.

References

[1] Y. Liu, T. Belkina, J. H. Hays, and R. Lublinerman, “Image de-fencing,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[2] Y. Mu, W. Liu, and S. Yan, “Video de-fencing,” *IEEE Trans. Circts. Sys. Vid. Tech.*, vol. 24, no. 7, pp. 1111–1121, 2014.

[3] T. Xue, M. Rubinstein, C. Liu, and W. T. Freeman, “A computational approach for obstruction-free photography,” *ACM Trans. Graph.*, vol. 34, no. 4, 2015.

[4] J. W. Renjiao Yi and P. Tan, “Automatic fence segmentation in videos of dynamic scenes,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[5] S. Jonna, K. K. Nakka, V. S. Khasare, R. R. Sahay, and M. S. Kankanhalli, “Detection and removal of fence occlusions in an image using a video of the static/dynamic scene,” *J. Opt. Soc. Am. A*, vol. 33, no. 10, pp. 1917–1930, 2016.

[6] M. Park, K. Brocklehurst, R. T. Collins, and Y. Liu, “Image de-fencing revisited,” in *Proc. Asian Conference on Computer vision*, 2010, pp. 422-434.

[7] V. S. Khasare, R. R. Sahay, and M. S. Kankanhalli, “Seeing through the fence: Image de-fencing using a video sequence,” in *Proc. IEEE Int. Conf. Image Process*, 2013, pp. 1351-1355.

[8] S. Jonna, K. K. Nakka, and R. R. Sahay, “My camera can see through fences: A deep learning approach for image de-fencing,” in *Proc. Asian Conference on Pattern Recognition*, 2015, pp. 261–265.

[9] A. Criminisi, P. Perez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200-1212, 2004.

[10] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *Int. J. Comput. Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[11] Z. Xiong, Y. Zhang, F. Wu, and W. Zeng, “Computational depth sensing: Toward high performance commodity depth cameras,” *IEEE Signal Processing Magazine*, vol. 34, 2017.

[12] A. V. Bhavsar and A. N. Rajagopalan, “Range map superresolution-inpainting, and reconstruction from sparse data,” *Computer Vision and Image Understanding*, vol. 116, no. 4, pp. 572–591, 2012.

[13] Lytro, Inc. https://www.lytro.com/.

[14] Raytrix, GmbH. https://www.raytrix.de/.

[15] H. G. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y. W. Tai, and I. S. Kweon, “Accurate depth map estimation from a lenslet light field camera,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1547–1555.

[16] M. W. Tao, P. P. Srinivasan, S. Hadap, S. Rusinkiewicz, J. Malik, and R. Ramamoorthi, “Shape estimation from

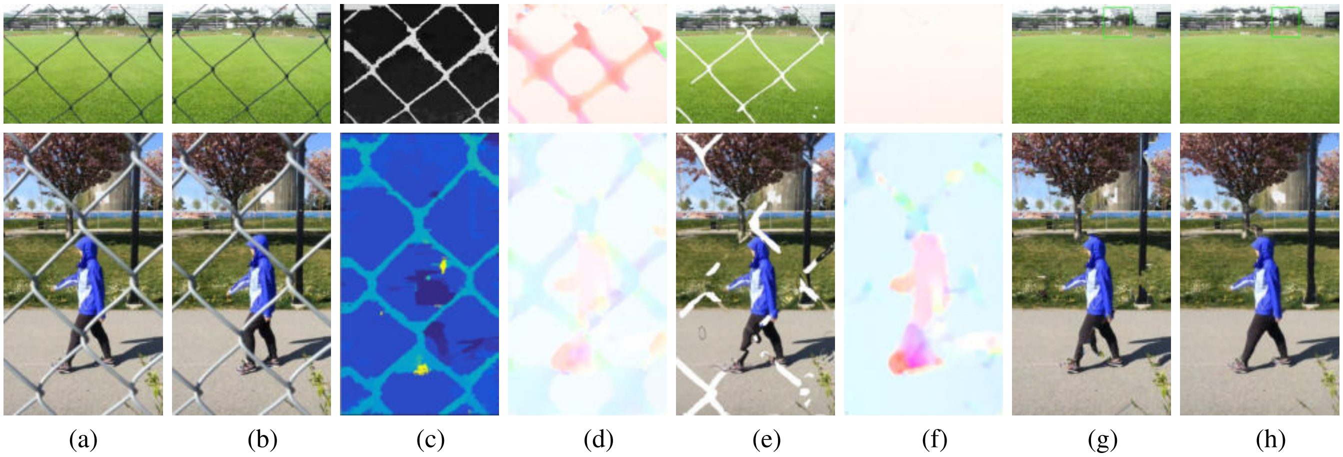


Fig. 14. **Results using stereo disparity maps**. Comparison with the state-of-the-art image de-fencing algorithm [2, 4]: (a), (b) Two frames taken from a video sequence reported in [2, 4]. (c) Disparity map obtained using the algorithm in [20]. (d) Optical flow estimated between the frames in (a), (b) using the algorithm [25]. (e) De-fenced image corresponding to (a) obtained using the flow in (d). (f) Occlusion-aware optical flow obtained using proposed algorithm. (g) Inpainted image obtained using [9] algorithm which was the technique followed in [4]. (h) De-fenced image obtained using proposed algorithm with occlusion-aware motion in (f).

shading, defocus, and correspondence using light-field angular coherence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 546–560, 2017.

[17] S. Z. Li, "Markov random field modeling in image analysis," *Springer*, 2001.

[18] D. Cho, S. Kim, Y. W. Tai, and I. S. Kweon, "Automatic trimap generation and consistent matting for light-field images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1504–1517, 2017.

[19] H. Mihara, T. Funatomi, K. Tanaka, H. Kubo, H. Nagahara, and Y. Mukaigawa, "4d light-field segmentation with spatial and angular consistencies," in *Proceedings of IEEE International Conference on Computational Photography (ICCP)*, 2016.

[20] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 504–511, 2013.

[21] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE MultiMedia*, vol. 19, no. 2, pp. 4–10, 2012.

[22] R. S. A. S. Raj, M. Lowney and G. Wetzstein, "Stanford lytro light field archive." [Online]. Available: http://lightfields.stanford.edu

[23] B. Zeisl, K. Koser, and M. Pollefeys, "Automatic registration of RGB-D scans via salient directions," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 2808–2815.

[24] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. ACM SIGGRAPH*, 2000, pp. 417-424.

[25] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, 2011.

[26] C. Liu, "Beyond pixels: Exploring new representations and applications for motion analysis," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.

[27] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, no. 1-4, pp. 259–268, 1992.

[28] T. Goldstein and S. Osher, "The split Bregman method for *l*1 regularized problems," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 323-343, 2009.

[29] S. Jonna, K. K. Nakka, and R. R. Sahay, "Deep learning based fence segmentation and removal from an image using a video sequence," in *Proc. European Conference on Computer Vision Workshops*, 2016, pp. 836–851.

[30] S. Jonna, S. Satapathy, and R. R. Sahay, "Stereo image de-fencing using smartphones," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 1792–1796.

[31] C. Guillemot and O. Le Meur, "Image inpainting : Overview and recent advances," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 127–144, 2014.

[32] J. Hays and A. A. Efros, "Scene completion using millions of photographs," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 1–7, 2007.

[33] Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1153-1165, 2010.

[34] K. A. Patwardhan, G. Sapiro, and M. Bertalmio, "Video inpainting under constrained camera motion," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 545-553, 2007.

[35] J. Gu, R. Ramamoorthi, P. Belhumeur, and S. Nayar, "Removing image artifacts due to dirty camera lenses and thin occluders," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 144:1–144:10, 2009.

[36] K. Papafitsoros, C. B. Schoenlieb, and B. Sengul, "Combined first and second order total variation inpainting using split Bregman," *Image Processing On Line*, vol. 3, pp. 112–136, 2013.

[37] K. He and J. Sun, "Image completion approaches using the statistics of similar patches," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2423-2435, 2014.

[38] T. Ruzic and A. Pizurica, "Context-aware patch-based image inpainting using Markov random field modeling," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 444–456, 2015.

[39] 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.*, vol. 24, no. 10, pp. 3034–3047, Oct 2015.

[40] J. S. Ren, L. Xu, Q. Yan, and W. Sun, "Shepard convolutional neural networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2015, pp. 901–909.

[41] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[42] A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proceedings of the International Conference on International Conference on Machine Learning*, 2016, pp. 1747–1756.

[43] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and Locally Consistent Image Completion," *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)*, vol. 36, no. 4, pp. 107:1–107:14, 2017.

[44] 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.*, vol. 31, no. 10, pp. 1804–1816, 2009.

[45] A. Yamashita, F. Tsurumi, T. Kaneko, and H. Asama, "Automatic removal of foreground occluder from multifocus images," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2012, pp. 5410–5416.

[46] Y. Zheng and C. Kambhamettu, "Learning based digital matting," in *Proc. Int. Conf. Comput. Vis.*, 2009.

[47] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, "Automatic inpainting by removing fence-like structures in RGBD images," *Machine Vision and Applications*, vol. 25, no. 7, pp. 1841–1858, 2014.

[48] L. Wang, H. Jin, R. Yang, and M. Gong, "Stereoscopic inpainting: Joint color and depth completion from stereo images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[49] J. Liu, X. Gong, and J. Liu, "Guided inpainting and filtering for kinect depth maps," in *Proceedings of the International Conference on Pattern Recognition*, 2012, pp. 2055–2058.

[50] F. Qi, J. Han, P. Wang, G. Shi, and F. Li, "Structure guided fusion for depth map inpainting," *Pattern Recognition Letters*, vol. 34, no. 1, pp. 70 – 76, 2013, extracting Semantics from Multi-Spectrum Video.

[51] S. Lu, X. Ren, and F. Liu, "Depth enhancement via low-rank matrix completion," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 3390–3397.

[52] H. Xue, S. Zhang, and D. Cai, "Depth image inpainting: Improving low rank matrix completion with low gradient regularization," *IEEE Trans. Image Processing*, vol. 26, no. 9, pp. 4311–4320, 2017.

[53] J. Park, H. Kim, Y. Tai, M. S. Brown, and I. Kweon, "High-quality depth map upsampling and completion for RGB-D cameras," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5559–5572, 2014.

[54] P. Buyssens, M. Daisy, D. Tschumperlé, and O. Lézoray, "Superpixel-based depth map inpainting for RGB-D view synthesis," in *International Conference on Image Processing*, 2015, pp. 4332–4336.

[55] W. Dong, G. Shi, X. Li, K. Peng, J. Wu, and Z. Guo, "Color-guided depth recovery via joint local structural and nonlocal low-rank regularization," *IEEE Trans. Multimedia*, vol. 19, no. 2, pp. 293–301, 2017.

[56] "Tango." [Online]. Available: https://get.google.com/tango/

[57] T.-C. Wang, J.-Y. Zhu, N. K. Kalantari, A. A. Efros, and R. Ramamoorthi, "Light field video capture using a learning-based hybrid imaging system," *ACM Transactions on Graphics*, vol. 36, no. 4, 2017.

[58] P. Guerrero, H. Winnemöller, W. Li, and N. J. Mitra, "Depthcut: Improved depth edge estimation using multiple unreliable channels," *CoRR*, vol. abs/1705.07844, 2017.

[59] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *ICLR*, 2016.

[60] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1744–1757, 2012.

[61] C. Liu, J. Yuen, and A. Torralba, "SIFT flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, 2011.

[62] C. Liu and D. Sun, "On bayesian adaptive video super resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 346–360, 2014.

[63] Q. Wang, S. Li, H. Qin, and A. Hao, "Super-resolution of multi-observed RGB-D images based on nonlocal regression and total variation," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1425–1440, 2016.

[64] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang, "Weighted nuclear norm minimization and its applications to low level vision," *International Journal of Computer Vision*, vol. 121, no. 2, pp. 183–208, 2017.

[65] S. Jonna, V. S. Voleti, R. R. Sahay, and M. S. Kankanhalli, "A multimodal approach for image de-fencing and depth inpainting," in *Proc. Int. Conf. Advances in Pattern Recognition*, 2015, pp. 1-6.