Diminishing Fence from Sweep Image Sequences Using Structure from Motion and Light Field Rendering

Chanya Lueangwattana
Keio University
Yokohama, Japan
chanyal@hvrl.ics.keio.ac.jp

Shohei Mori
Keio University
Yokohama, Japan
mori@hvrl.ics.keio.ac.jp

Hideo Saito
Keio University
Yokohama, Japan
saito@hvrl.ics.keio.ac.jp

ABSTRACT

Diminishing the appearance of a fence in an image, so-called *de-fencing*, is a challenging research area due to the characteristics of fences (i.e., thin, texture-less, etc.) and a requirement for occluded background restoration. In this paper, we describe a de-fencing method for an image sequence captured via a user's sweep motion, in which occluded background is potentially observed. To make use of geometric and appearance natures of such consecutive images, we use two well-known approaches: Structure from motion and light field rendering. The results using real image sequences showed that our method is superior to an image-inpainting-based approach in some use cases.

ACM Classification Keywords

H.5.1. Information Interfaces and Presentation: Multimedia Information Systems; Artificial, augmented, and virtual realities

Author Keywords

Image de-fencing; diminished reality; structure from motion; light field rendering

INTRODUCTION

Recovering occluded objects in a scene, also known as diminished reality (DR), is a challenging issue that has recently received increasing attention [12]. Visual obstacles are seen through with variety techniques such as image-inpainting, which is used to fill in the obstacle pixels with ones having similar features. Due to the similar problem statements, we can consider image *de-fencing* a kind of DR method. Defencing refers to techniques to diminish the appearance of a fence in an image to create a fence-free view. Such a technique is useful, for example, when a photographer takes a photo of a tourist landmark but the scene is occluded by fences for security reasons.

There are two challenging issues in de-fencing; 1) fence detection and 2) background restoration (i.e., restoration of pixels on fence pixels).

Fence detection: The difficulties in fence detection and segmentation lays in the characteristics of fences; they are thin, texture-less, etc. Thus, many existing methods of segmenting fence pixels from other pixels require fully manual or semi-automated efforts [13], although recent work has introduced some automated methods [3, 4, 5, 9, 8].

Background restoration: To fill in the detected fence pixels, we can use one of DR methods, image-inpainting, to fill in the pixels with the other pixels [8], or multi-viewpoint images to observe the hidden regions [5, 9].

An original image de-fencing work was proposed by Liu *et al*. [8], which introduced foreground-background segmentation based on the fact that fences have nearly regular patterns. Whereas, the method has limitation due to the algorithm relying on ideal regular patterns and uses a single image resource for background restoration. To overcome such limitations, later researches used richer resources, videos, for these tasks since occluded regions in a frame are visible in the other frames. [5, 9] are pioneering works that used videos for the defencing problem. They identified fence pixels by differences in optical flow at each frame. [3, 4] used depth information since fences always appear closer to the camera than the background.

As an alternative of these methods, we propose combining a well-known computer vision and a graphics method, structure from motion (SfM) and light field rendering (LFR) respectively. Our contributions can be summarized as follows:

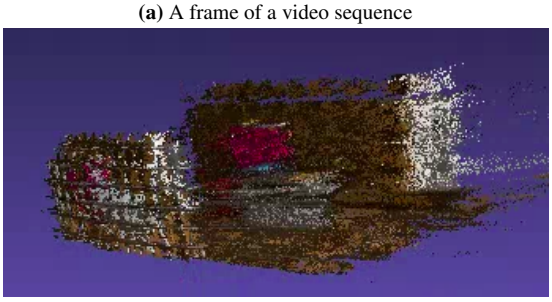
- Segmenting foreground and background regions based on depth information from SfM and dense reconstruction,
- Recovering occluded background pixels using a modified LFR scheme assuming an image sequence captured by user's sweep camera motion,
- and a framework for combining the above two approaches for de-fencing.

DE-FENCING USING SFM AND LFR

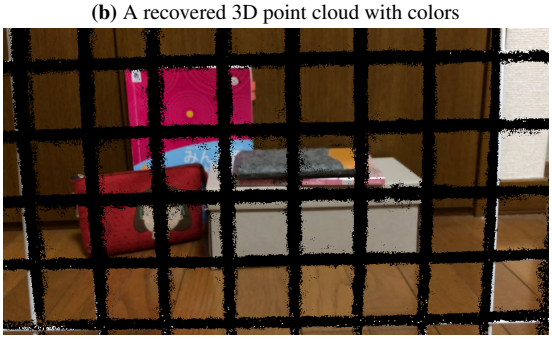
Scene Capture

Following the literatures [5, 9], we expect that regions occluded by fences in a frame are observable in other frames in a

(a) A frame of a video sequence



(b) A recovered 3D point cloud with colors



(c) A frame with fence mask obtained by back-projection of the 3D point cloud of fence structure

Figure 1. Fence masking using a 3D point cloud

video. To record an image sequence, we have to move the camera in a diagonal direction against the fence rectangle. Note that this diagonal sweep motion is essential for the proposed method to make the camera fully observes the background.

Fence Detection

Geometric information of the captured scene is recovered as a 3D point cloud by SfM and multi-view stereo [10, 11] using COLMAP, followed by separation of fence and non-fence point clouds. Figure 1 (a) and (b) shows an example of an input frame and the corresponding 3D point cloud respectively. The feature points in each frame are detected by SIFT and matched among the consecutive frames using sequential matching. From obtained point correspondences, each frame is registered with its camera pose and triangulated points as a sparse point cloud. Then, depth and normal map are computed from registered pairs, and fused to the sparse point cloud to reconstruct dense one. Note that all frames share the same intrinsic parameters given by bundle adjustment in the SfM since the video is captured using a single camera.

After the dense 3D reconstruction, we separate the 3D point cloud to fence and non-fence ones. To achieve this, we obtain $T\%$ closest points (e.g., $T = 35$) among the 3D point cloud in a camera coordinate system as a fence point cloud. Figure 1 (c) shows a frame with fence mask, which is a re-projection of such fence point cloud colored in black. This re-projection is computed by perspective transformation with camera parameters and a camera pose that extracted from each frame in SfM phase, as in the following equation.

$$\sigma\tilde{\mathbf{x}} = \mathbf{A}[\mathbf{R}|\mathbf{t}]\tilde{\mathbf{X}} \tag{1}$$

where σ is a scale factor, $\tilde{\mathbf{x}}$ is the homogeneous re-projected point coordinates, $\tilde{\mathbf{X}}$ is homogeneous 3D point coordinates, $\mathbf{A}$ is a 3×3 matrix of intrinsic parameters, and $[\mathbf{R}|\mathbf{t}]$ is the 3×4 matrix of extrinsic parameters describing the camera motion.

Background Restoration

Here, we recover the missing pixels in the detected fence regions based on LFR, which is an image based rendering method for generating new views from arbitrary camera positions [1, 2, 6, 7]. LFR uses four parameters, $\mathbf{r} = (u, v, s, t)$, to represent a scene. As shown in Figure 2, a ray $\mathbf{r}$ represents a light ray that passes through a camera plane at (u, v) and a focal plane at (s, t) in a virtual image C. A pixel color at (s, t) in the visual view C is, therefore, calculated by blending the corresponding colors in data cameras' images D_i.

To make use of the obtained data so far, we modified the LFR [1, 2] as described in the pseudo code in Algorithm 1. Given a background 3D point $\mathbf{X}$ at a missing pixel position $\mathbf{x}$ and registered data cameras D_i, we render the background by blending the D_i images. The pixels to be blended in D_i are calculated by projecting the 3D point to D_i. However, the masked pixels in D_i are given zero weight for the blending and, as a result, the fence pixels are not counted for the blending. In addition, the blending function is weighted with inverse proportional of euclidean distance from (u, v)'s camera position to the render frame position, which means more weight is given to a ray from the camera that aligns closer to C.

EXPERIMENTAL RESULTS

Setup

We set up five experimental scenes with combinations of various types of objects and fences to confirm the robustness of the proposed method against scene variations. Figure 3 (a) shows the scenes including indoor (Scene 1, 2, and 3) and outdoor scenes (Scene 4 and 5). We compare results by the proposed method and by PhotoShop Content-Aware Fill (i.e., image-inpainting) as a baseline. Note that both of the methods use the same mask images as in Figure 3 (b) given by an approach described in Section 2.2. Here, our discussions are limited to qualitative manners and quantitative ones remain in our future work.

We recorded five image sequences in the scenes using iPhone 8 Plus (960×540 pixels at 30Hz). For each scene, we recorded 399, 519, 499, 534, and 463 frames of videos and used 40, 40, 40, 54, and 180 closest data cameras in Euclidean distance for the LFR. On the other hand, the baseline method used a single

Algorithm 1: LFR for De-fencing

$I^C(\mathbf{x})$: Color at $\mathbf{x} \in \mathbb{R}^2$ of the user camera C
$I^{D_i}(\mathbf{x})$: Color at $\mathbf{x}$ of a data camera D_i
$I_M^{D_i}(\mathbf{x})$: $\{0, 1\}$ at $\mathbf{x}$ of a data camera D_i
$\mathbf{X}$: 3D position corresponding to $I^C(\mathbf{x})$ (i.e., focal plane)
$I^R(\mathbf{x})$: Resultant color of de-fenced LFR at $\mathbf{x}$

foreach D_i **do**
  $d_i \leftarrow EuclideanDistance(C, D_i)$
foreach $\mathbf{x}$ *within* I^C **do**
  $sum_{d_i} \leftarrow 0$
  foreach D_i **do**
    /* Ψ projects input 3D point to a camera */
    $\mathbf{x}' \leftarrow \Psi(\mathbf{X}, \mathbf{D_i})$
    $sum_{d_i} \leftarrow sum_{d_i} + I_M^{D_i}(\mathbf{x}') \frac{1}{exp(d_i^2)}$
  $I^R(\mathbf{x}) \leftarrow black$
  foreach D_i **do**
    $\mathbf{x}' \leftarrow \Psi(\mathbf{X}, \mathbf{D_i})$
    $I^R(\mathbf{x}) \leftarrow I^R(\mathbf{x}) + I_M^{D_i}(\mathbf{x}') \frac{I^{D_i}(\mathbf{x}')}{exp(d_i^2)}$
  $I^R(\mathbf{x}) \leftarrow \frac{I^R(\mathbf{x})}{sum_{d_i}}$

image as a resource for its background restoration (i.e., Figure 3 (a)).

De-fencing Results and Discussions

Figure 3 (c) and (d) shows the results of the proposed method and the baseline. In most scenes, the proposed method gives us impressions that the fences are diminished. However, we should note that the de-fencing by the proposed method replaces the masked pixels with the background pixels observed at different frames while the region results in blurry artifacts. Especially in Scene 4 and 5 where the fence regions relatively densely exist, the proposed method obtains distorted images. We consider that such blurry effects can be reduced by using depth information given by the dense reconstruction, while we currently assume that the focal plane is placed at infinity (i.e., $\mathbf{X}$ is at infinity).

On the other hand, the baseline method does not produce such blurry effects, while it tends to recover fence pixels due to the remaining fence pixels. That is, we consider that the remaining fence pixels induced by inaccurate fence masking gives cues for image-inpainting scheme to optimize to fence restoration rather than background restoration.

CONCLUSION

In this paper, we proposed an alternative method for image de-fencing, which gives a framework of a combination of SfM-based fence detection and LFR-based background restoration. The qualitative evaluations showed that the proposed method suffered from a trade-off between the number of blended data

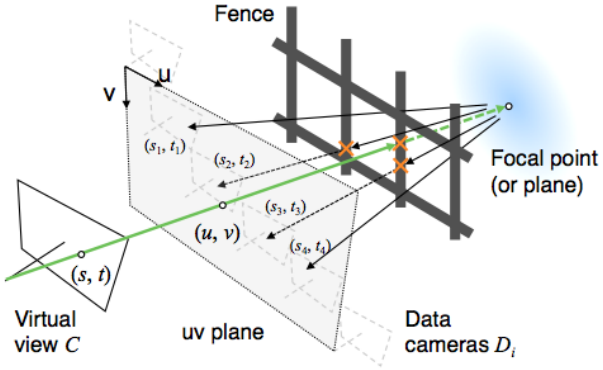


Figure 2. Light field rendering parameterization

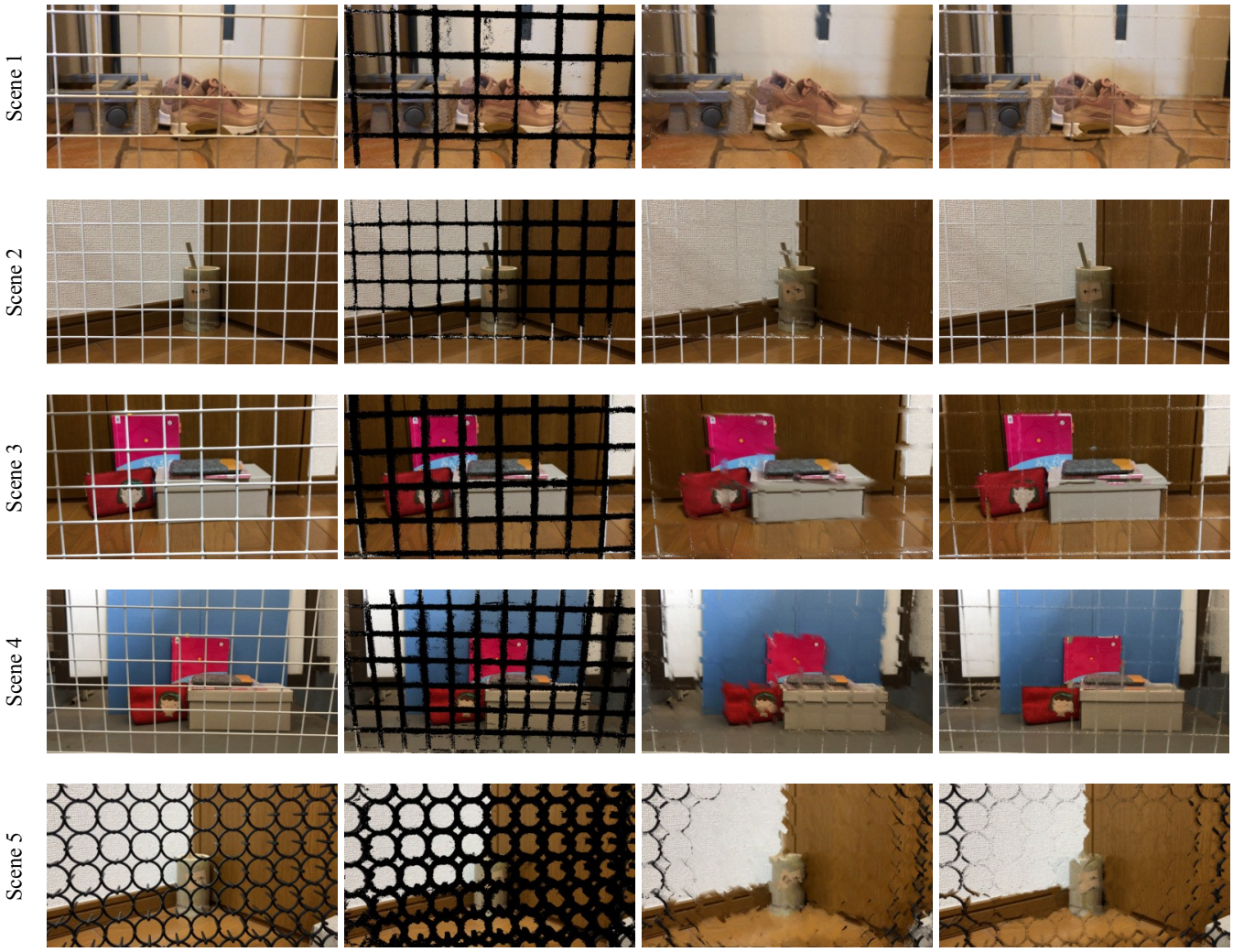
camera images and blurry artifacts at image edges. However, we also suggested that the proposed method gives more reasonable background information than the baseline image-inpainting method does due to its multi-viewpoint image nature. In the future work, we will extend our evaluations to quantitative manners and use SfM-induced depth information for reducing blurry effects in the results.

ACKNOWLEDGEMENT

This work was supported in part by a Grant-in-Aid from the Japan Society for the Promotion of Science Fellows Grant Number 16J05114.

REFERENCES

1. A. Davis, M. Levoy, and F. Durand. 2012. Unstructured light fields. *Computer Graphics Forum* 31, 2pt1 (2012).
2. A. Isaksen, L. Mcmillan, and S. J. Gortler. 2000. Dynamically reparameterized light fields. In *Proc. Int. Conf. and Exhibition on Computer Graphics and Interactive Technique (SIGGRAPH)*.
3. S. Jonna, S. Satapathy, and R. R. Sahay. 2017. Stereo image de-fencing using smartphones. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. 1792–1796.
4. S. Jonna, V. S. Voleti, R. R. Sahay, and M. S. Kankanhalli. 2015. A multimodal approach for image de-fencing and depth inpainting. In *Proc. Int. Conf. on Advances in Pattern Recognition (ICAPR)*.
5. V. S. Khasare, R. R. Sahay, and M. S. Kankanhalli. 2013. Seeing through the fence: Image de-fencing using a video sequence. In *Proc. IEEE Int. Conf. on Image Processing (ICIP)*.
6. N. Kusumoto, S. Hiura, and K. Sato. 2009. Uncalibrated synthetic aperture for defocus control. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2552–2559.
7. M. Levoy and P. Hanrahan. 1996. Light field rendering. In *Proc. Int. Conf. and Exhibition on Computer Graphics and Interactive Technique (SIGGRAPH)*.



(a) An input frame (b) Fence-masked (a) (c) De-fencing result (ours) (d) De-fencing result (Photoshop)

Figure 3. Experimental results

8. Y. Liu, T. Belkina, J. H. Hays, and R. Lublinerman. 2008. Image de-fencing. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
9. M. Park, K. Brocklehurst, R. T. Collins, and Y. Liu. 2011. Image de-fencing revisited. In *Proc. Asian Conf. on Computer Vision (ACCV), Lecture Notes in Computer Science*. 422–434.
10. J. L. Schönberger and J.-M. Frahm. 2016. Structure-from-motion revisited. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
11. J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. 2016. Pixelwise view selection for unstructured multi-view stereo. In *Proc. European Conf. on Computer Vision (ECCV)*.
12. S. Shohei, S. Ikeda, and H. Saito. 2017. A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects. *IPSJ Trans. on Computer Vision and Applications (CVA)* 9, 17 (2017). DOI:http://dx.doi.org/10.1186/s41074-017-0028-1
13. Y. Zheng and C. Kambhamettu. 2009. Learning based digital matting. In *Proc. IEEE Conf. on Computer Vision (ICCV)*.