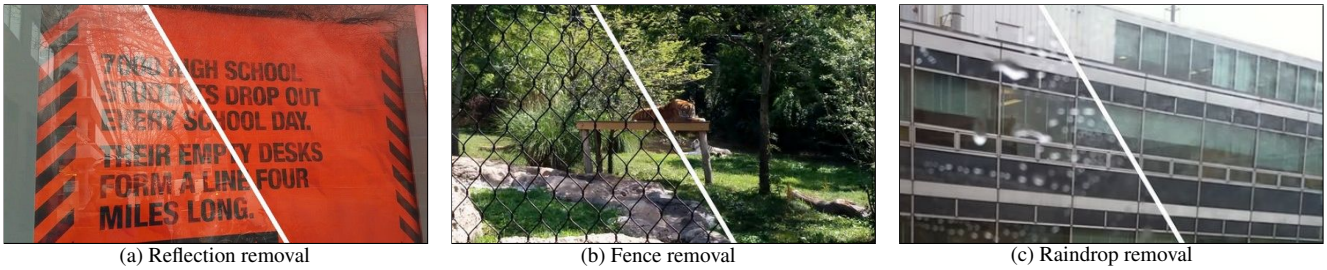
Learning to See Through Obstructions

Yu-Lun Liu[1,4] Wei-Sheng Lai[2] Ming-Hsuan Yang[2,3] Yung-Yu Chuang[1] Jia-Bin Huang[5]
[1]National Taiwan University [2]Google [3]UC Merced [4]MediaTek Inc. [5]Virginia Tech
https://www.cmlab.csie.ntu.edu.tw/~yulunliu/ObstructionRemoval



(a) Reflection removal (b) Fence removal (c) Raindrop removal

Figure 1: **Seeing through obstructions.** We present a learning-based method for recovering clean images from a given short sequence of images taken by a moving camera through obstructing elements such as (a) windows, (b) fence, or (c) raindrop.

Abstract

We present a learning-based approach for removing unwanted obstructions, such as window reflections, fence occlusions or raindrops, from a short sequence of images captured by a moving camera. Our method leverages the motion differences between the background and the obstructing elements to recover both layers. Specifically, we alternate between estimating dense optical flow fields of the two layers and reconstructing each layer from the flow-warped images via a deep convolutional neural network. The learning-based layer reconstruction allows us to accommodate potential errors in the flow estimation and brittle assumptions such as brightness consistency. We show that training on synthetically generated data transfers well to real images. Our results on numerous challenging scenarios of reflection and fence removal demonstrate the effectiveness of the proposed method.

1. Introduction

Taking clean photographs through reflective surfaces (such as windows) or occluding elements (such as fences) is challenging as the captured images inevitably contain both the scenes of interests and the obstructions caused by reflections or occlusions. An effective solution to recover the underlying clean image is thus of great interest for improving the quality of the images captured under such conditions or allowing computers to form a correct physical interpretation of the scene, e.g., enabling a robot to navigate in a scene with windows safely.

Recent efforts have been focused on automatically removing unwanted reflections or occlusions from *a single image* [2, 8, 16, 17, 27, 38, 43, 45]. These methods either leverage the ghosting cues [30] or adopt learning-based approaches to capture the prior of natural images [8, 16, 38, 43, 45]. While impressive results have been shown, separating the clean background from reflection/occlusions is fundamentally ill-posed and often requires a high-level semantic understanding of the scene to perform well. In particular, the performance of learning-based methods degrades significantly for out-of-distribution images.

To tackle these challenges, *multi-frame approaches* have been proposed for reflection/occlusion removal. The core idea is to exploit the fact that the background scene and the occluding elements are located at different depths with respect to the camera (e.g., virtual depth of window reflections). Consequently, taking multiple images from a slightly moving camera reveals the motion differences between the two layers [3, 9, 12, 21, 24, 34]. A number of approaches exploit such cues for reflection or fence removal from a video [1, 3, 6, 9, 12, 21, 24, 26, 31, 34]. Xue et al. [42] propose a unified computational framework for obstruction removal and show impressive results on several natural sequences. The formulation, however, requires a computationally expensive optimization process and relies on strict assumptions of brightness constancy or accurate motion estimation. To alleviate these issues, recent work [1] explores model-free methods by using a generic 3D convolutional neural network (CNN). Yet, the CNN-based methods do not produce results with comparable quality as optimization-based algorithms on real input sequences.

In this work, we propose a multi-frame obstruction removal algorithm that exploits the advantages of both optimization-based and learning-based methods. Inspired by the optimization-based approach [42], the proposed algorithm alternates between the dense motion estimation and the background/obstruction layer reconstruction steps in a coarse-to-fine manner. The explicit modeling of dense motion allows us to progressively recover detailed content in the respective layers. Instead of relying on hand-crafted objectives for solving the layers, we exploit the learning-based method for fusing flow-warped images to accommodate potential violations of brightness constancy and errors in flow estimation. We train our fusion network using a synthetically generated dataset and demonstrate it transfers well to unseen real-world sequences. In addition, we present an online optimization process to further improve the visual quality of particular testing sequences. Finally, we demonstrate that the proposed method performs favorably against existing algorithms on a wide variety of challenging sequences and applications.

Our framework builds upon the optimization-based formulation of [26, 42] but differs in that our model is purely data-driven and does not rely on classical assumptions such as brightness constancy [26, 42], accurate flow fields [21], or planar surface [12] in the scene. When these assumptions are violated (e.g., occlusion/dis-occlusion, motion blur, inaccurate flow), classical approaches may fail to reconstruct clear foreground and background layers. On the other hand, data-driven approaches learn from diverse training data and can tolerate errors when these assumptions are violated.

The contributions of this work include:

- We present a learning-based method that integrates the optimization-based formulation for robustly reconstructing background/obstruction layers.
- We demonstrate that combining model pre-training using synthetically generated data and fine-tuning with real testing sequence (in an unsupervised manner) leads to state-of-the-art performance.
- We show our model with minimum design changes can be applied to various obstruction removal problems.

2. Related work

Multi-frame reflection removal. Existing methods often exploit the differences of motion patterns between the background and reflection layers [12, 42] and impose natural image priors [10, 12, 42]. These methods differ in their way of modeling the motion fields, e.g., SIFT flow [21], homography [12], and dense optical flow [42]. Recent advances include optimizing temporal coherence [26] and learning-based layer decomposition [1]. Compared to learning a generic CNN [1], our method explicitly models the dense flow fields of the background and obstruction layers to obtain sharper and cleaner results on real sequences.

Single-image reflection removal. A number of approaches have been proposed to remove unwanted reflections with only *one single image* as input. Existing methods exploit various cues, including ghosting effect [30], blurriness caused by depth-of-field [22, 36], image priors (either hand-designed [2] or learned from data [43, 45]), and the defocus-disparity cues from dual pixel sensors [28]. Despite the demonstrated success, reflection removal from a single image remains challenging due to the nature of this highly ill-posed problem and the lack of motion cues. Our work instead utilizes the motion cues from image sequences captured with a slightly moving camera for separating the background and reflection layers.

Occlusion and fence removal. Occlusion removal aims to eliminate the captured obstructions, e.g., fence or raindrops on an image or sequences, and provide a clear view of the scene. Existing methods detect fence patterns by exploiting visual parallax [25], dense flow field [42], disparity maps [18], or using a graph-cut [44]. One recent work leverages a CNN for fence segmentation [6] and recovers the occluded pixels using optical flow. Our method also learns deep CNNs for optical flow estimation and background image reconstruction. Instead of focusing on fence removal, our formulation is more general and applicable to different obstruction removal tasks.

Video completion. Video completion aims to fill in plausible content in missing regions of a video [14], with applications ranging from object removal, full-frame video stabilization, and watermark/transcript removal. State-of-the-art methods estimate the flow fields in both known and missing regions to constrain the content synthesis [13, 40], and generate temporally coherent results. The obstruction removal problem resembles a video completion task. However, the crucial difference is that no manual mask selection is required for removing the fences/obstructions from videos.

Layer decomposition. Image layer decomposition is a long-standing problem in computer vision, e.g., intrinsic image [4, 46], depth, normal estimation [15], relighting [7], and inverse rendering [23, 29]. Our method is inspired by the development of the approaches for these layer decomposition, particularly in the ways of leveraging both the physical image formation constraints and data-driven priors.

Online optimization. Learning from the test data has been an effective way to reduce the domain discrepancy between the training/testing distributions. Examples include using geometric constraints [5], self-supervised losses [33], and online template update [19]. Similar to these methods, we apply online optimization to fine-tune our background/obstruction reconstruction network on a particular test sequence to further improve the separation. Our unsupervised loss directly measures how well the recovered background/obstruction and the dense flow fields explain all the input frames.

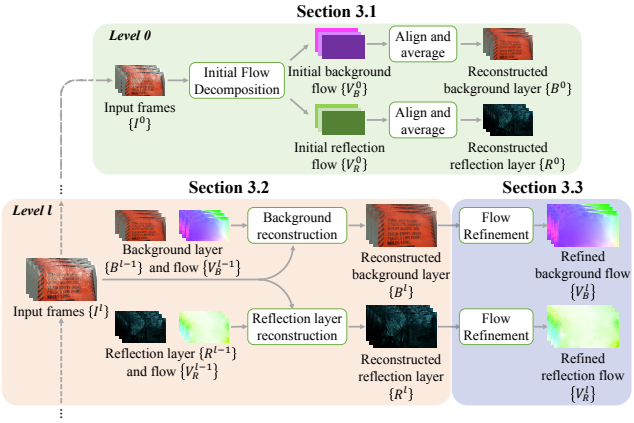


Figure 2: **Algorithmic overview.** We reconstruct the background/reflection layers in a coarse-to-fine manner. At the coarsest level, we estimate uniform flow fields for both the background and reflection layers and then reconstruct coarse background/reflection layers by averaging the aligned frames. At level l, we apply (1) background/reflection layer reconstruction modules to reconstruct the background/reflection layer, and (2) use the PWC-Net to predict the refined flow fields for both layers. Our framework progressively reconstructs the background/reflection layers and flow fields until the finest level.

3. Proposed Method

Given a sequence $\{I_t\}_{t=1}^T$ of T frames, the goal is to decompose each frame I_k into two layers, one for the (clean) background and the other for obstruction caused by fense/raindrops/occlusion. Decomposing an image sequence into background and obstruction layers is difficult as it involves solving two tightly coupled problems: optical flow decomposition and layer reconstruction. Without a good flow decomposition, the layers cannot be reconstructed faithfully due to the misalignment from inaccurate motion estimation. On the other hand, without well-reconstructed background and obstruction layers, the optical flow cannot be accurately estimated because of the mixed content. Due to the nature of this chicken-and-egg problem, there is no ground to start with because we do not have information for both flows and layers.

In this work, we propose to learn deep CNNs to address the challenges. Our proposed method mainly consists of three modules: 1) initial flow decomposition, 2) background and obstruction layer reconstruction, and 3) optical flow refinement. Our method takes T frames as input and aims to decompose the keyframe frame I_k into a background layer B_k and reflection layer R_k at a time. We reconstruct the output images in a coarse-to-fine manner within an L-level hierarchy. First, we estimate the flows at the coarsest level from the initial flow decomposition module (Section 3.1). We then progressively reconstruct the background/obstruction layers (Section 3.2) and refine optical flows (Section 3.3) until the last level. Figure 2 shows an overview of our method. Our unified framework can be applied to several layer decomposition problems, such as reflection/obstruction/fence/rain removal. Without loss of generality, we use the reflection removal task as an example to introduce our algorithm. We describe the details of the three modules in the following sections.

3.1. Initial Flow Decomposition

We first predict the flow for both background and reflection layers at the coarsest level ($l = 0$), which is the essential starting point of our algorithm. Instead of estimating dense flow fields, we propose to learn a *uniform* motion vector for each layer. Our initial flow decomposition network consists of two sub-modules: 1) a feature extractor, and 2) a layer flow estimator. The feature extractor first generates feature maps for all the input frames at a $1/2^L\times$ spatial resolution. Then, we construct a cost volume between frame j and frame k via a correlation layer [32]:

$$CV_{jk}(\mathbf{x_1}, \mathbf{x_2}) = c_j(\mathbf{x_1})^\top c_k(\mathbf{x_2}), \tag{1}$$

where c_j and c_k are the extracted features of frame j and k, respectively, and $\mathbf{x}$ indicates the pixel index. Since the spatial resolution is quite small at this level, we set the search range of the correlation layer to only 4 pixels. The cost volume CV is then concatenated with the feature c_j and fed into the layer flow estimator.

The layer flow estimator uses the global average pooling and fully-connected layers to generate two global motion vectors. Finally, we tile the global motion vectors into two uniform flow fields (at a $1/2^L\times$ spatial resolution): $\{V^0_{B,j\to k}\}$ for the background layer and $\{V^0_{R,j\to k}\}$ for the reflection layer. We provide the detailed architecture of our initial flow decomposition module in the supplementary material.

3.2. Background/Reflection Layer Reconstruction

The layer reconstruction module aims to reconstruct the clean background image B_k and the reflection image R_k. Although the two tasks of background and reflection reconstruction are similar in their goals, the characteristics of the background and reflection layers are quite different. For example, the background layers are often more dominant in appearance but could be occluded in some frames. On the other hand, the reflection layers are often blurry and darker. Consequently, we train two independent networks for reconstructing the background and reflection layers. The two networks have the same architecture but do not share the network parameters. In the following, we only describe the network for background layer reconstruction; the reflection layer is reconstructed in a similar fashion.

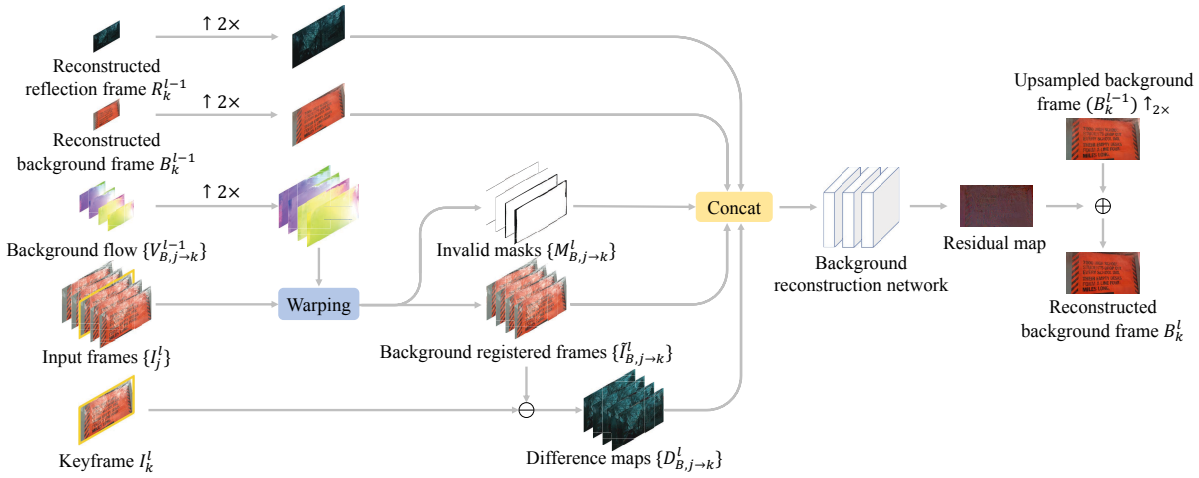


Figure 3: **Overview of layer reconstruction module.** At level l, we first upsample the background flows $\{V_{B,j\to k}^{l-1}\}$ from level $l-1$ to warp and align the input frames $\{I_t^l\}$ with the keyframe I_k^l. We then compute the difference maps between the background-registered frames and the keyframe. The background reconstruction network takes as input the background-registered frames $\{\tilde{I}_{B,j\to k}^l\}$, the difference maps $\{D_{B,j\to k}^l\}$, the invalid masks $\{M_{B,j\to k}^l\}$, the upsampled background $(B_k^{l-1})\uparrow_2$, the reflection layers $(R_k^{l-1})\uparrow_2$, and learns to predict the residual map of the background keyframe. We add the predicted residual map to the upsampled background frame $(B_k^{l-1})\uparrow_2$ and produce the reconstructed background frame B_k^l at level l. For the reflection layer reconstruction, we use the same architecture but learn a different set of network parameters.

We reconstruct the background layer in a coarse-to-fine fashion. At the coarsest level ($l=0$), we first use the flow fields estimated from the initial flow decomposition module to align the neighboring frames. Then, we compute the average of all the background-registered frames as the predicted background image:

$$B_k^0 = \frac{1}{T}\sum_{j=1}^{T} \boldsymbol{W}(I_j^0, V_{B,j\to k}^0), \tag{2}$$

where I_j^0 is the frame j downsampled to level 0, and $\boldsymbol{W}()$ is the bilinear sampling operation.

At the l-th level, the network takes as input the reconstructed background image B_k^{l-1}, reflection image R_k^{l-1}, background optical flows $\{V_{B,k\to j}^{l-1}\}$ from the previous level as well as the input frames $\{I_t^l\}$ at the current level. The model aims to reconstruct the background image of the keyframe B_k^l at the current level. We first upsample the background flow fields $\{V_{B,k\to j}^{l-1}\}$ by $2\times$ and align all the input frames $\{I_j^l\}$ to the keyframe $\{I_k^l\}$:

$$\tilde{I}_{B,j\to k}^l = \boldsymbol{W}(I_j^l, (V_{B,j\to k}^{l-1})\uparrow_2), \tag{3}$$

where $()\uparrow_2$ denotes the $2\times$ bilinear upsampling operator. As some pixels may become invalid due to occlusion or the warping from outside image boundaries, we also compute a difference map $D_{B,j\to k}^l = |I_{B,j\to k}^l - I_k^l|$ and a warping invalid masks $M_{B,j\to k}^l$ as additional cues for the network to reduce the warping artifacts.

We concatenate the registered frames, difference maps, invalid masks, and the upsampled background and reflection layers from the previous level as the input feature to the background reconstruction network. The network then reconstructs a background image B_k^l via residual learning:

$$B_k^l = g_B\Big(\{\tilde{I}_{B,j\to k}^l\}, \{D_{B,j\to k}^l\}, \{M_{B,j\to k}^l\}, (B_k^{l-1})\uparrow_2, (R_k^{l-1})\uparrow_2\Big) + (B_k^{l-1})\uparrow_2, \tag{4}$$

where g_B is the background reconstruction network. Note that the reflection layer is also involved in the reconstruction of the background layer, which couples the background and reflection reconstruction networks together for joint training. Figure 3 illustrates an overview of the background reconstruction network at the l-th level. The detailed network configuration is provided in the supplementary material.

3.3. Optical Flow Refinement

After reconstructing all the background images B^l, we then learn to refine the background optical flows. We use the pre-trained PWC-Net [32] to estimate the flow fields between a paired of background images:

$$V_{B,j\to k}^l = \text{PWC}(B_j^l, B_k^l), \tag{5}$$

where PWC is the pre-trained PWC-Net. Note that the PWC-Net is fixed and not updated with the other sub-modules of our model.

3.4. Network Training

To improve training stability, we employ a two-stage training procedure. At the first stage, we train the initial flow decomposition network with the following loss:

$$\mathcal{L}_{\text{dec}} = \sum_{k=1}^{T} \sum_{j=1, j\neq k}^{T} \|V^0_{B,j\to k} - \text{PWC}(\hat{B}_j, \hat{B}_k)\downarrow^{2^L}\|_1 + \|V^0_{R,j\to k} - \text{PWC}(\hat{R}_j, \hat{R}_k)\downarrow^{2^L}\|_1 , \tag{6}$$

where $\downarrow$ is the bilinear downsampling operator, $\hat{B}$ and $\hat{R}$ denote the ground-truth background and reflection layers, respectively. We use the pre-trained PWC-Net to compute optical flows and downsample the flows by $2^L\times$ as the ground-truth to train the initial flow decomposition network.

Next, we freeze the initial flow decomposition network and train the layer reconstruction networks with an image reconstruction loss:

$$\mathcal{L}_{\text{img}} = \frac{1}{T\times L}\sum_{t=1}^{T}\sum_{l=0}^{L}(\|\hat{B}^l_t - B^l_t\|_1 + \|\hat{R}^l_t - R^l_t\|_1), \tag{7}$$

and a gradient loss:

$$\mathcal{L}_{\text{grad}} = \frac{1}{T\times L}\sum_{t=1}^{T}\sum_{l=0}^{L}(\|\nabla\hat{B}^l_t - \nabla B^l_t\|_1 + \|\nabla\hat{R}^l_t - \nabla R^l_t\|_1), \tag{8}$$

where ∇ is the spatial gradient operator. The gradient loss encourages the network to reconstruct faithful edges to further improve visual quality. The overall loss for training the layer reconstruction networks is:

$$\mathcal{L} = \mathcal{L}_{\text{img}} + \lambda_{grad}\mathcal{L}_{\text{grad}}, \tag{9}$$

where the weight λ_{grad} is empirically set to 1 in all our experiments. We train both the initial flow decomposition and layer reconstruction networks with the Adam optimizer [20] with a batch size of 2. We set the learning rate to 10^{-4} for the first 100k iterations and then decrease to 10^{-5} for another 100k iterations.

3.5. Synthetic Sequence Generation

Since collecting real sequences with ground-truth reflection and background layers is very difficult, we use the Vimeo-90k dataset [41] to synthesize sequences for training. Out of the 91,701 sequences in the Vimeo-90k training set, we randomly select two sequences as the background and reflection layers. First, we warp the sequences using random homography transformations. We then randomly crop the sequences to a spatial resolution of 320×192 pixels. Finally, the composition is applied frame by frame using the realistic reflection image synthesis model proposed by previous work [8, 45]. More details about the synthetic data generation are provided in the supplementary material.

3.6. Online Optimization

We observe that the model trained on our synthetic dataset may not perform well on real-world sequences. Therefore, we propose an online refinement method to fine-tune our pre-trained model with real sequences by optimizing an unsupervised warping consistency loss:

$$\mathcal{L}_{\text{warp}} = \sum_{k=1}^{T}\sum_{j=0, j\neq k}^{T}\sum_{l=0}^{L}\|I^l_j - (\boldsymbol{W}(B^l_k, V^l_{B,j\to k}) + \boldsymbol{W}(R^l_k, V^l_{R,j\to k}))\|_1. \tag{10}$$

The consistency loss enhances fidelity by enforcing that the predicted background and reflection layers should be warped back and composited into the original input frames. In addition, we also incorporate the total variation loss:

$$\mathcal{L}_{tv} = \sum_{t=1}^{T}\sum_{l=0}^{L}(\|\nabla B^l_t\|_1 + \|\nabla R^l_t\|_1), \tag{11}$$

which encourages the network to generate natural images by following the sparse gradient image prior. The overall loss of online optimization is:

$$\mathcal{L}_{online} = \mathcal{L}_{\text{warp}} + \lambda_{tv}\mathcal{L}_{tv}, \tag{12}$$

where the weight λ_{tv} is empirically set to 0.1 in all our experiments. Note that we freeze the weight of the PWC-Net and only update the background/reflection layer reconstruction modules. We fine-tune our model on every single input sequence for 1k iterations, which takes about 20 minutes for a sequence with a 1296×864 spatial resolution. We use only five frames in the sequence for fine-tuning.

3.7. Extension to Other Obstruction Removal

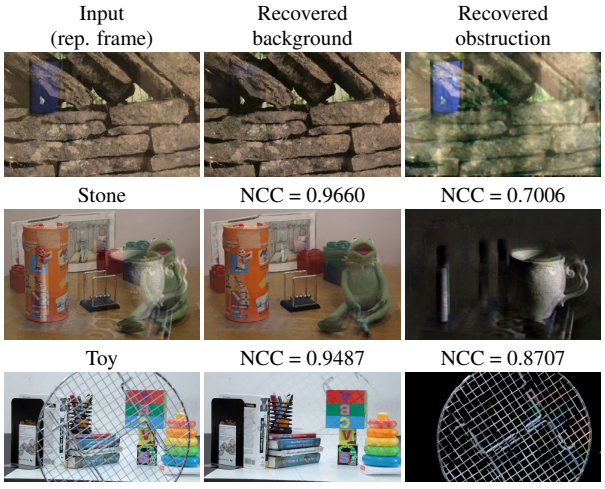
The proposed framework can be easily modified to handle other obstruction removal tasks, such as fence or raindrop removal. First, we remove the image reconstruction network for the obstruction (i.e., reflection) layer and only predict the background layers. Second, the background image reconstruction network outputs an additional channel as the alpha map for segmenting the obstruction layer. We do not estimate flow fields for the obstruction layer as the flow estimation network cannot handle the repetitive structures (e.g., fence) or tiny objects (e.g., raindrops) well and often predicts noisy flows. With such a design change, our model is able to perform well on the fence and raindrop removal tasks. We use the fence segmentation dataset [6] and alpha matting dataset [39] to train our model for both tasks.

4. Experiments and Analysis

We present the main findings in this section and include more results in the supplementary material.

Table 1: **Quantitative comparison of reflection removal methods on synthetic sequences.** We compare the proposed method with existing reflection removal approaches on a synthetic dataset with 100 sequences, where each sequence contains five consecutive frames. For the single-image based methods [8, 16, 38, 43, 45], we generate the results frame-by-frame. For multi-frame algorithms [1, 12, 21] and our method, we use five input frames to generate the results.

Method			Background				Reflection			
			PSNR ↑	SSIM ↑	NCC ↑	LMSE ↓	PSNR ↑	SSIM ↑	NCC ↑	LMSE ↓
Single image	CEILNet [8]	CNN-based	20.35	0.7429	0.8547	0.0277	-	-	-	-
	Zhang et al. [45]	CNN-based	19.53	0.7584	0.8526	0.0207	18.69	0.4945	0.6283	0.1108
	BDN [43]	CNN-based	17.08	0.7163	0.7669	0.0288	-	-	-	-
	ERRNet [38]	CNN-based	22.42	0.8192	0.8759	0.0177	-	-	-	-
	Jin et al. [16]	CNN-based	18.65	0.7597	0.7872	0.0218	11.44	0.3607	0.4606	0.1150
Multiple images	Li and Brown [21]	Optimization-based	17.12	0.6367	0.6673	0.0604	7.68	0.2670	0.3490	0.1214
	Guo et al. [12]	Optimization-based	14.58	0.5077	0.5802	0.0694	14.12	0.3150	0.3516	0.1774
	Alayrac et al. [1]	CNN-based	23.62	0.7867	0.9023	0.0200	21.18	0.6320	0.7535	0.1517
	Ours w/o online optim.	CNN-based	**26.57**	**0.8676**	**0.9380**	**0.0125**	**21.42**	**0.6438**	**0.7613**	**0.1008**



Method	Stone		Toy		Hanoi	
	B	R	B	R	B	O
Li and Brown [21]	0.9271	0.2423	0.7906	0.6084	-	-
Guo et al. [12]	0.7258	0.1018	0.7701	0.6860	-	-
Xue et al. [42]	**0.9738**	**0.8433**	0.8985	0.7536	0.9921	0.7079
Alayrac et al. [1]	0.9367	0.1633	0.7985	0.5263	-	-
Ours	0.9660	0.7006	**0.9487**	**0.8707**	**0.9938**	**0.8267**

Figure 4: **Quantitative evaluation on controlled sequences.** For each sequence, we show the keyframe (*left*) and recovered background (*middle*) and reflection/occluder (*right*). We report the NCC scores of recovered backgrounds and reflections for quantitative comparisons.

4.1. Comparisons with State-of-the-arts

Controlled sequences. We first evaluate on the controlled sequences provided by Xue et al [42], which contain three videos with ground-truth background and reflection layers. We compare the proposed method with Li and Brown [21], Guo et al. [12], Xue et al. [42], and Alayrac et al. [1]. Figure 4 shows our recovered background and reflection/obstruction layers and the normalized cross-correlation (NCC) scores [35, 42]. Our method performs favorably against other approaches on the Toy and Hanoi sequences and shows comparable scores to Xue et al. [42] on the Stone sequence.

Synthetic sequences. We synthesize 100 sequences by the method described in Section 3.5 from the Vimeo-90k test set. We compare our approach with five single-image reflection removal methods [8, 16, 38, 43, 45], and three multi-frame approaches [1, 12, 21]. We use the default parameters of each method to generate the results. Since Alayrac et al. [1] do not release the source code or pre-trained model, we re-implement their model and train on our training dataset. Table 1 shows the average PSNR, SSIM [37], NCC, and LMSE [11] metrics. The proposed method obtains the best scores on all the evaluation metrics for both background and reflection layers.

Real sequences. In Figure 5, we present visual comparisons of real input sequences from [42]. Our method is able to separate the reflection layers and reconstruct clear and sharp background images than other approaches [1, 21, 26, 42]. Figure 6 shows two examples where the inputs contain obstruction such as texts on the glass or raindrops. Our method can remove the obstruction layer and reconstruct clear background images. More visual comparisons are available in the supplementary material.

4.2. Analysis and Discussion

In this section, we analyze several key design choices of the proposed framework. We also provide the execution time and show a failure case of our method.

Initial flow decomposition. We demonstrate that the uniform flow initialization plays an important role in our algorithm. We train our model with the following settings: 1) removing the initial flow decomposition network, where the flows at the coarsest level are set to zero, and 2) predicting spatially-varying dense flow fields as the initial flows.

Figure 5: **Visual comparison of background-reflection separation on natural sequences.** More results can be found in the supplementary material. *Results are in lower resolution.

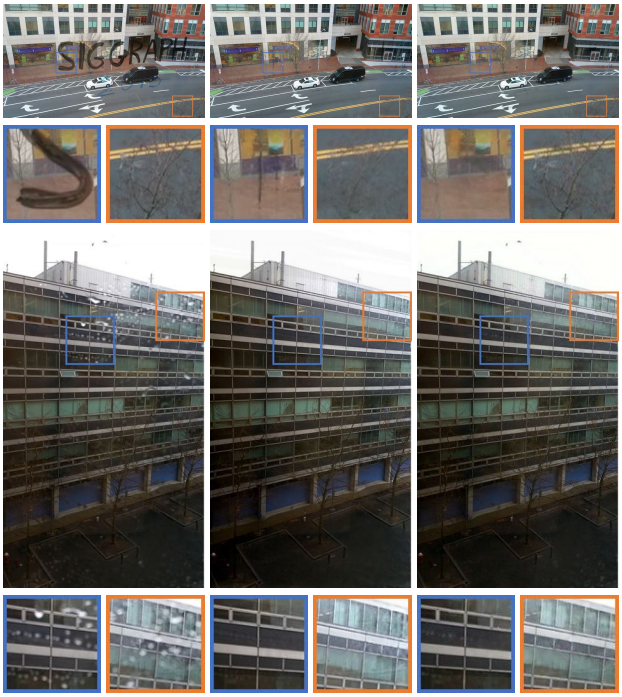


Figure 6: **Recovering occluded scenes by raindops.**

Table 2(a) reports the validation loss of Equation (9) on our Vimeo-90k validation set, where the model with uniform flow prediction achieves a much lower validation loss compared to the alternatives. Initializing the flow fields to zero makes it difficult for the following levels to decompose the background and reflection layers. On the contrary, estimating dense flow fields at the coarsest level may result in noisy predictions and lead to inconsistent layer separation. Our uniform flow prediction strikes a balance and serves as a good initial prediction to facilitate the following background reconstruction and flow refinement steps.

Image reconstruction network. To demonstrate the effectiveness of the image reconstruction network, we replace it with a temporal filter to fuse the neighbor frames, which are warped and aligned by the optical flows. We show in Table 2(b) that both the temporal mean and median filters result in large errors (in terms of the validation loss of Equation (9)) as the errors are accumulated across levels. In contrast, our image reconstruction network learns to reduce warping and alignment errors and generates clean foreground and background images.

Online optimization. Table 2(c) shows that both the network pre-training with synthetic data and online optimization with real data are beneficial to the performance of our model. In Figure 7, we show that the model without pre-training cannot separate the reflection well on the real input sequence. Without online optimization, the background image contains residuals from the reflection layer. After online optimization, our method is able to reconstruct both background and reflection layers well.

Running time. We evaluate the execution time of two optimization-based algorithms [12, 21] and a recent CNN-based method [1] with different input sequences resolutions on a computer with Intel Core i7-8550U CPU and NVIDIA TITAN Xp GPU. Table 3 shows that our method without the online optimization step runs faster than optimization-based algorithms. Alayrac et al. [1] use a 3D CNN architecture without explicit motion estimation, which results in a faster inference speed. In contrast, our method computes bi-directional optical flows for every pair of input frames in

Figure 7: **Effect of online optimization and pre-training.** Both steps are crucial to achieving high-quality results.

Table 2: **Ablations.** We analyze the design choices of the proposed method and report the validation loss of Equation (9) on the synthetic reflection-background Vimeo-90k test set.

(a) **Initial flow decomposition**: Predicting uniform flow fields as initialization achieves better results.

Flow initialization	Loss
Zero initialization	0.377
Dense flow field	0.226
Uniform flow field (Ours)	**0.184**

(b) **Fusion method**: Our image reconstruction network recovers better background/reflection than temporal mean/median filtering.

Image fusion method	Loss
Temporal mean filtering	0.526
Temporal median filtering	0.482
Image reconstruction network (Ours)	**0.184**

(c) **Model training**: Both the network pre-training and online optimization are important to the performance of our method.

Online optimization	Pre-training	Loss
✓	-	0.417
-	✓	0.184
✓	✓	**0.139**

Table 3: **Running time comparison (in seconds).** CPU: Intel Core i7-8550U, GPU: NVIDIA TITAN Xp. * denotes methods using GPU.

	QVGA (320×240)	VGA (640×480)	720p (1280×720)
Li and Brown [21]	82.591	388.235	1304.231
Guo et al. [12]	64.251	369.200	1129.125
*Alayrac et al. [1]	0.549	2.011	6.327
*Ours w/o online optim.	1.107	2.216	9.857
*Ours w/ online optim.	66.056	264.227	929.182

Figure 8: **A failure case.** Our method fails to recover the correct flow fields for each layer, leading to ineffective reflection removal.

a coarse-to-fine manner, which is slower but achieves much better reconstruction performance.

Failure case. We show a failure case of our algorithm in Figure 8, where our method does not separate the reflection layer well. This example is particularly challenging as there are two layers of reflections: the top part contains the wooden beams, and the bottom part comes from the street behind the camera. As the motion of the wooden beams is close to the background image, our method can only separate the street scenes in the reflection layer.

5. Conclusions

We have presented a novel method for multi-frame reflections and obstructions removal. Our key insight is to leverage a CNN to reconstruct background and reflection layers from flow-warped images. Integrating optical flow estimation and coarse-to-fine refinement enable our model to robustly recover the underlying clean image from challenging real-world sequences. Our method can be applied to different tasks such as fence or raindrop removal with minimum changes in our design. We also show that online optimization on testing sequences leads to improved visual quality. Extensive visual comparisons and quantitative evaluation demonstrate that our approach performs well on a wide variety of scenes.

Acknowledgments. This work is supported in part by NSF CAREER (#1149783), NSF CRII (#1755785), MOST 109-2634-F-002-032, MediaTek Inc. and gifts from Adobe, Toyota, Panasonic, Samsung, NEC, Verisk, and Nvidia.

References

[1] Jean-Baptiste Alayrac, Joao Carreira, and Andrew Zisserman. The visual centrifuge: Model-free layered video representations. In *CVPR*, 2019. 1, 2, 6, 7, 8

[2] Nikolaos Arvanitopoulos, Radhakrishna Achanta, and Sabine Susstrunk. Single image reflection suppression. In *CVPR*, 2017. 1, 2

[3] Efrat Be'Ery and Arie Yeredor. Blind separation of superimposed shifted images using parameterized joint diagonalization. *TIP*, 17(3):340–353, 2008. 1

[4] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *ACM TOG*, 33(4):159, 2014. 2

[5] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *ICCV*, 2019. 2

[6] Chen Du, Byeongkeun Kang, Zheng Xu, Ji Dai, and Truong Nguyen. Accurate and efficient video de-fencing using convolutional neural networks and temporal information. In *ICME*, 2018. 1, 2, 5

[7] Elmar Eisemann and Frédo Durand. Flash photography enhancement via intrinsic relighting. *ACM TOG*, 23(3):673–678, 2004. 2

[8] Qingnan Fan, Jiaolong Yang, Gang Hua, Baoquan Chen, and David Wipf. A generic deep architecture for single image reflection removal and image smoothing. In *ICCV*, 2017. 1, 5, 6

[9] Kun Gai, Zhenwei Shi, and Changshui Zhang. Blind separation of superimposed images with unknown motions. In *CVPR*, 2009. 1

[10] Kun Gai, Zhenwei Shi, and Changshui Zhang. Blind separation of superimposed moving images using image statistics. *TPAMI*, 34(1):19–32, 2011. 2

[11] Roger Grosse, Micah K Johnson, Edward H Adelson, and William T Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *ICCV*, 2009. 6

[12] Xiaojie Guo, Xiaochun Cao, and Yi Ma. Robust separation of reflection from multiple images. In *CVPR*, 2014. 1, 2, 6, 7, 8

[13] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Temporally coherent completion of dynamic video. *ACM TOG*, 35(6):196, 2016. 2

[14] Shachar Ilan and Ariel Shamir. A survey on data-driven video completion. *Computer Graphics Forum*, 34(6):60–85, 2015. 2

[15] Junho Jeon, Sunghyun Cho, Xin Tong, and Seungyong Lee. Intrinsic image decomposition using structure-texture separation and surface normals. In *ECCV*, 2014. 2

[16] Meiguang Jin, Sabine Süsstrunk, and Paolo Favaro. Learning to see through reflections. In *ICCP*, 2018. 1, 6

[17] Sankaraganesh Jonna, Krishna K Nakka, and Rajiv R Sahay. Deep learning based fence segmentation and removal from an image using a video sequence. In *ECCV*, 2016. 1

[18] Sankaraganesh Jonna, Sukla Satapathy, and Rajiv R Sahay. Stereo image de-fencing using smartphones. In *ICASSP*, 2017. 2

[19] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *TPAMI*, 34(7):1409–1422, 2011. 2

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[21] Yu Li and Michael S Brown. Exploiting reflection change for automatic reflection removal. In *ICCV*, 2013. 1, 2, 6, 7, 8

[22] Yu Li and Michael S Brown. Single image layer separation using relative smoothness. In *CVPR*, 2014. 2

[23] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *CVPR*, 2020. 2

[24] Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic, and William T Freeman. Sift flow: Dense correspondence across different scenes. In *ECCV*, 2008. 1

[25] Yadong Mu, Wei Liu, and Shuicheng Yan. Video de-fencing. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(7):1111–1121, 2013. 2

[26] Ajay Nandoriya, Mohamed Elgharib, Changil Kim, Mohamed Hefeeda, and Wojciech Matusik. Video reflection removal through spatio-temporal optimization. In *ICCV*, 2017. 1, 2, 6, 7

[27] Minwoo Park, Kyle Brocklehurst, Robert T Collins, and Yanxi Liu. Image de-fencing revisited. In *ACCV*, 2010. 1

[28] Abhijith Punnappurath and Michael S Brown. Reflection removal using a dual-pixel sensor. In *CVPR*, 2019. 2

[29] Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David W Jacobs, and Jan Kautz. Neural inverse rendering of an indoor scene from a single image. In *ICCV*, 2019. 2

[30] YiChang Shih, Dilip Krishnan, Fredo Durand, and William T Freeman. Reflection removal using ghosting cues. In *CVPR*, 2015. 1, 2

[31] Sudipta N Sinha, Johannes Kopf, Michael Goesele, Daniel Scharstein, and Richard Szeliski. Image-based rendering for scenes with reflections. *ACM TOG*, 31(4):100–1, 2012. 1

[32] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 3, 4

[33] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A Efros, and Moritz Hardt. Test-time training for out-of-distribution generalization. *arXiv:1909.13231*, 2019. 2

[34] Richard Szeliski, Shai Avidan, and P Anandan. Layer extraction from multiple images containing reflections and transparency. In *CVPR*, 2000. 1

[35] Renjie Wan, Boxin Shi, Ling-Yu Duan, Ah-Hwee Tan, and Alex C Kot. Benchmarking single-image reflection removal algorithms. In *ICCV*, 2017. 6

[36] Renjie Wan, Boxin Shi, Tan Ah Hwee, and Alex C Kot. Depth of field guided reflection removal. In *ICIP*, 2016. 2

[37] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *TIP*, 13(4):600–612, 2004. 6

[38] Kaixuan Wei, Jiaolong Yang, Ying Fu, David Wipf, and Hua Huang. Single image reflection removal exploiting misaligned training data and network enhancements. In *CVPR*, 2019. 1, 6

[39] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *CVPR*, 2017. 5

[40] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. In *CVPR*, 2019. 2

[41] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *IJCV*, 127(8):1106–1125, 2019. 5

[42] Tianfan Xue, Michael Rubinstein, Ce Liu, and William T Freeman. A computational approach for obstruction-free photography. *ACM TOG*, 34(4):79, 2015. 1, 2, 6, 7

[43] Jie Yang, Dong Gong, Lingqiao Liu, and Qinfeng Shi. Seeing deeply and bidirectionally: A deep learning approach for single image reflection removal. In *ECCV*, 2018. 1, 2, 6

[44] Renjiao Yi, Jue Wang, and Ping Tan. Automatic fence segmentation in videos of dynamic scenes. In *CVPR*, 2016. 2

[45] Xuaner Zhang, Ren Ng, and Qifeng Chen. Single image reflection separation with perceptual losses. In *CVPR*, 2018. 1, 2, 5, 6

[46] Tinghui Zhou, Philipp Krahenbuhl, and Alexei A Efros. Learning data-driven reflectance priors for intrinsic image decomposition. In *ICCV*, 2015. 2