Video Reflection Removal Through Spatio-Temporal Optimization

Ajay Nandoriya*[1], Mohamed Elgharib*[1], Changil Kim[2], Mohamed Hefeeda[3], and Wojciech Matusik[2]

[1]Qatar Computing Research Institute, HBKU [2]MIT CSAIL [3]Simon Fraser University

Abstract

Reflections can obstruct content during video capture and hence their removal is desirable. Current removal techniques are designed for still images, extracting only one reflection (foreground) and one background layer from the input. When extended to videos, unpleasant artifacts such as temporal flickering and incomplete separation are generated. We present a technique for video reflection removal by jointly solving for motion and separation. The novelty of our work is in our optimization formulation as well as the motion initialization strategy. We present a novel spatio-temporal optimization that takes n frames as input and directly estimates $2n$ frames as output, n for each layer. We aim to fully utilize spatio-temporal information in our objective terms. Our motion initialization is based on iterative frame-to-frame alignment instead of the direct alignment used by current approaches. We compare against advanced video extensions of the state of the art, and we significantly reduce temporal flickering and improve separation. In addition, we reduce image blur and recover moving objects more accurately. We validate our approach through subjective and objective evaluations on real and controlled data.

1. Introduction

The popularity of digital videography is driven by the continuous rise of mobile imaging, whether being an expensive stand-alone camera or a commodity camera embedded in a phone. Such devices are usually coupled with computational photography software to enhance the quality of photographs beyond the physical and hardware limitations [1]. One situation where such enhancement is desirable is in reflection handling. Reflections can obstruct the original scene (see Fig. 1), and hence their removal can be desirable.

Optics-driven separation techniques use polarized filters. This, however, requires careful manual tweaking, generates incomplete removal and dims the overall illumination. A number of computational separation approaches are proposed. However, all are developed to extract one reflection and one background image for the entire input. Some techniques take a single still image as input [14, 13, 20], some take image pairs [18, 29] and others take a sequence of images [27, 28, 19, 22, 10]. Techniques using sequence of images exploit the observation that separation can be generated through temporal filtering. This requires stabilizing the examined layer beforehand. This category of techniques commonly solves jointly for motion and layers [27, 22, 19, 28, 6].

No current technique is designed to extract videos from the input. Current approaches for doing so is to warp the single-image results through time [28]. This, however, lacks any moving object and often looks artificial. An alternative approach is applying separation techniques frame by frame followed by temporal filtering. This, however, produces strong temporal flickering and incomplete removal. It also blurs moving objects.

We present a technique for video reflection removal. The novelty of our work lies in both our optimization function as well as the motion initialization strategy. Our optimization takes n frames as input and directly estimates $2n$ frames as output, n for each layer. We aim to fully utilize spatio-temporal information in our optimization terms (see Tab. 1). Motions are initialized through iterative frame-to-frame instead of direct alignment used in state of the art. Here, we use a combination of feature tracks and edge-flow. This improves separation quality. Our approach is iterative, continuously refining separation and motion. Comparing against advanced video extensions of the state of the art [28, 29], our technique significantly reduces temporal flickering and improves the quality of reflection removal (see Fig. 1). It also reduces blurring of moving objects.

2. State of the Art

A mixed image I is modeled as a linear combination of the original background B and the foreground reflection F : $I = \alpha B + \beta F$, where (α, β) are mixing parameters. A

*Indicates equal contributions. Contact us through: ahmedelm@tcd.ie

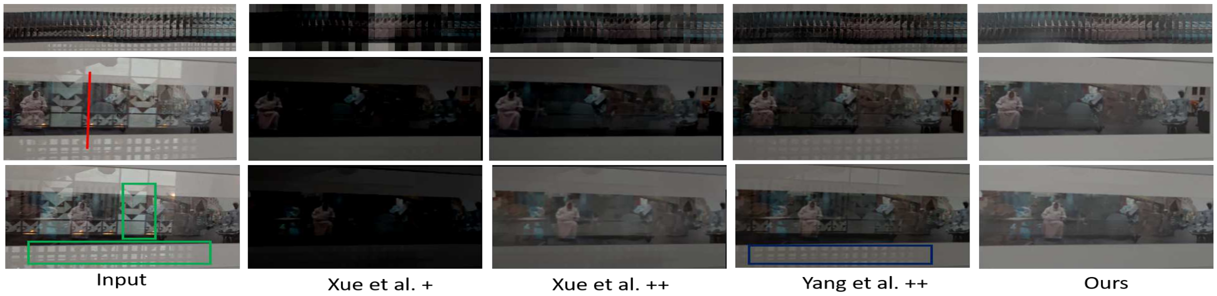

Figure 1. Video reflection removal with different techniques. The reflection is in the form of a bright pattern (see green). The first row zooms in on the spatio-temporal red slice area and concatenates it through time. The current removal techniques are designed for still-images. Extending them to videos produces strong temporal flickering and poor separation (see red slices and blue region). Our approach, however, generates temporally coherent results with more complete removal. We used our more advanced alignment strategy for all methods.

number of techniques have been proposed for separating I into B and F [27, 28, 22, 29, 18, 20, 13, 15]. Since it is a severely ill-posed problem, more constraints are required. Levin et al. [14, 13] observed that high frequency spatial components such as edges and corners are sparse. The separation of reflection is generated by maximizing the sparsity of such features in the estimated layers. Their technique, however, requires tense manual identification of the high frequency components. Shih et al. [20] handles only reflections containing repetitive ghosting, common in thick glasses. Another class of techniques requires two mixtures of B and F under two different mixing proportions (α_1, β_1) and (α_2, β_2) [18, 2, 8]. Sarel et al. [18] minimize the grayscale correlation between the underlying layers through iterative information exchange. Farid et al. [8] use independent component analysis (ICA) to achieve the separation. Similarly, Bronstein et al. [2] use ICA but on a sparse representation such as edges and corners.

Motion is a commonly used cue for reflection separation [27, 22, 19, 28, 10, 15, 9]. B and F are aligned in time and separation is achieved by temporal filtering. Alignment is done through direct frame-to-frame matching. Solution is refined iteratively by solving for separation and motion [27, 22, 19, 28, 10]. Szeliski et al. [22] recover B and F through a minimum and maximum temporal filtering. Weiss et al. [27] recover a single intrinsic image (B) through a temporal median operator. Sarel et al. [19] assume repetitive F motion and exploit that for stabilization. F is then separated through Weiss et al.'s [27] temporal filtering. Li et al. [15] build on Levin et al.'s [13] single-image decomposition and removes manual interaction by exploiting motion information. SIFT-flow [17] is used to directly warp all frames over a reference. Edge labeling is found using per pixel spatio-temporal variation. Here, stable gradients are treated as background. Levin et al. [13] is applied for each frame and the final background is taken as the minimum of all separated backgrounds. Both Guo et al. [10] and Gai et al. [9] take multiple frames as input but produce one frame as output. Their technique is based on global parametric motion modeling and does not handle local movements.

	Xue et al. [28]	Yang et al. [29]	Ours
Image compositing	✓	✓	✓
Temporal layer redundancy	✗	✓	✓
Layer prior	✓	✗	✓
Iterative alignment	✗	✗	✓

Table 1. *Objective terms and alignment strategy of different techniques. We aim to fully utilize spatio-temporal information.*

Xue et al. [28] and Yang et al. [29] are the latest related techniques. Yang et al. focus on optical flow estimation between two frames while Xue et al. focus on layer separation. Yang et al. initialize solution using Li et al. [15] and optimize for layer temporal redundancy, image compositing, intensity and motion smoothness. Their optimization does not have a layer prior term. Xue et al.'s use edge-flow to generate two homographies, one for each layer. A dense motion field is computed for both layers, followed by a temporal stabilization. The background is initialized through Szeliski et al.'s [22] minimum temporal filtering and the foreground is taken as the residual. Xue et al. optimization has an image compositing term and an intensity smoothness term. However, unlike Yang et al., they have a layer prior term. This improves separation quality. Tab. 1 summarizes the different objective terms used by different techniques.

Others addressed reflections in different context. Kopf et al. [11] and Sinha et al. [21] propose techniques for image-based rendering of reflections. Tsin et al. [23] addressed stereo matching for reflections. Elgharib et al. [5, 7] detects reflections in videos. Lang et al. [12] extends still image techniques to video. Their technique, however, can not handle the two layer assumption of reflections.

3. Our Approach

A mixed video is modeled as a linear combination of the background B and the reflection F using $I_t = B_t + F_t$.

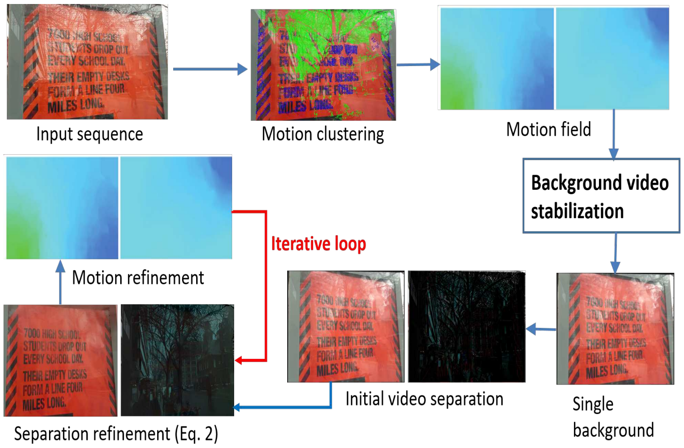

Figure 2. Algorithm overview. We estimate and segment motions into two clusters. We assume background have the most features. The video is warped over one background layer and reflection removed through temporal minimum filtering. Video is generated by warping the background and reflection is taken as the remaining component. Layers are refined through our spatio-temporal optimization (Eq. 2) followed by optical flow for motion refinement. The process is repeated for better results.

Here, t denotes the frame number. We assume (α, β) are temporally constant. Exploiting the observation that video frames can be warped on another, the compositing equation becomes

$$I_t = W^F_{t,\rho} \cdot F_\rho + W^B_{t,\rho} \cdot B_\rho \qquad (1)$$

$W^F_{t,\rho}$ is the F motion field from frame ρ to t. We solve for F_t and B_t and their motion field $W^F_{t,\rho}$ and $W^B_{t,\rho}$. Our approach directly estimates n B and n F frames. This imposes temporal coherency and maintains image sharpness. We use feature tracks and an iterative edge-flow matching for initialization. Our technique is iterative, where each iteration refines motion and reflection removal.

3.1. Objective Function

Our objective function is composed of three terms

$$E = \lambda_d E_d + \lambda_l E_l + \lambda_s E_s. \qquad (2)$$

E_d is the data term to ensure the recovered layers satisfy the image compositing model and that they can be warped from other points in time (Eq. 1). E_l is a layer prior defining to which layer each edge of the input I belongs to. E_s is spatial smoothness, while $(\lambda_d, \lambda_l, \lambda_s)$ are hyper-parameters to configure the importance of each term. They are fixed to $(2, 2, 1)$ in all experiments.

The data term minimize the difference between a layer at time t and its warped version from time ρ through $E'_d(\rho) = \sum_{t=1}^{N} \left(\|B_t - W^B_{t,\rho} \cdot B_\rho\|_1 + \|F_t - W^F_{t,\rho} \cdot F_\rho\|_1\right)$. Here, $\|.\|_1$ is the L1-norm. N is the total number of frames of the examined sequence. Instead of biasing the solution towards a specific reference, we generalize ρ to include all frames and hence modify the data term to $E''_d(\rho) = \sum_{\rho=1}^{N} \sum_{t=1}^{N} \left(\|B_t - W^B_{t,\rho} \cdot B_\rho\|_1 + \|F_t - W^F_{t,\rho} \cdot F_\rho\|_1\right)$. We substitute the reflection ($F_t = I_t - B_t$) and obtain

$$E_d = \sum_{\rho=1}^{N} \sum_{t=1}^{N} \left(\|B_t - W^B_{t,\rho} \cdot B_\rho\|_1 + \|(I_t - B_t) - W^F_{t,\rho} \cdot (I_\rho - B_\rho)\|_1\right). \qquad (3)$$

The layer prior imposes labeling constraints on the separated layers and is defined through M_t. M_t is a binary map defining to what layer each I pixel belongs to (0 for B). M_t is estimated by thresholding the alignment errors of the high frequency components (see Sec. 3.3). Layer estimates are then constrained by minimizing

$$E_l = \sum_{t=1}^{N} \left(M_t \nabla I_t \cdot |\nabla B_t| + (1 - M_t) \nabla I_t \cdot |\nabla (I_t - B_t)|\right). \qquad (4)$$

Here, ∇I_t is estimated by a Canny edge detector and $|\nabla B_t|$ is the first order spatial gradient of B_t. Finally, we impose spatial smoothness on the reconstructed layers through $E_s = \sum_{t=1}^{N} \left(|\nabla B_t| + |\nabla (I_t - B_t)|\right)$. This is done by minimizing the first order spatial gradient.

Fig. 2 shows an overview of our algorithm. Motion are separated into two clusters. The green and blue dots stick to F and B respectively. Homography for each layer is estimated and sequence is stabilized over B. Background is initialized through a minimum temporal filtering and warped back to the remaining frames. Reflections are taken as the residual component through $F = I - B$. Separation is improved by Eq. 2. We iteratively refine our motion and layer estimates until the convergence is reached.

3.2. Motion Initialization

The first main step in our algorithm is estimating the warping fields $W_{t,\rho}$ for each layer (Eq. 3). Given the input sequence I, our aim is to register it over frame t using either F or B motion

$$I^S_t(\mathbf{x}, \rho) := I(W_{t,\rho}(\mathbf{x}), \rho), \qquad (5)$$

Here, $\mathbf{x}$ denotes 2D pixel co-ordinates. Given a set of points X_t in frame t and their corresponding X_ρ in the reference, we find $W_{t,\rho}$ by minimizing $\|W_{t,\rho}(X_\rho) - X_t\|^2_2$. Xue et al. [28] used direct edge-flow matching to define the feature correspondence. Such direct matching can be erroneous especially as the gap between t and ρ increases. This generates temporally incoherent results, which is problematic for videos. Instead, we propose two different approaches for motion initialization. We first attempt to use feature point tracks [25]. If no enough tracks are available then we use iterative edge-flow matching.

Feature point tracks [25] provide direct correspondence between frame t and the reference ρ. This makes them attractive for our application. Given that, we estimate $W_{t,\rho}$

directly. Edge-flow is effective for capturing weak structures [17]. We define our warping $W_{t,\rho}$ as a combination of two terms: one direct $W^d_{t,\rho}$ and one iterative $W^i_{t,\rho}$

$$W_{t,\rho} = \lambda_d W^d_{t,\rho} + \lambda_i W^i_{t,\rho}, \tag{6}$$

(λ_d, λ_i) are weights to configure the importance of each term. They are fixed to $(\lambda_d, \lambda_i) = (1, 5)$ in all experiments. If tracks are used instead, then we do not need to configure (λ_d, λ_i). The direct warping $W^d_{t,\rho}$ is estimated in a similar manner to tracks. The iterative warping term, however, is different.

For a set of points in frame t and their correspondence in the next frame, we match them with one homography. We impose temporal smoothness on W using a moving average filter. We use a local window of 5 frames centered on the examined frame t and weighted by $\omega \sim \mathcal{N}(0, 4)$. We use a temporally iterative scheme to estimate W between the examined frame t and the reference ρ. First, we generate an estimate of W for each pair of consecutive frames. For instance, if $\rho > t$ we estimate $W_{t+1,t}, W_{t+2,t+1}, W_{t+3,t+2}, ...W_{\rho,\rho-1}$. Hence, the direct transformation from t to ρ becomes $W_{t,\rho} = \prod_{u=t}^{u=\rho-1} W_{u+1,u}$. For $\rho < t$, we do the same process but in the opposite time direction.

3.3. Separation Initialization and Refinement

We group frame correspondences into two segments using k-means clustering, and fit one homography for each. The background is stabilized and separated using minimum temporal filtering. We warp the initial background to all frames and estimate the corresponding N reflections. Layer labeling mask M_t of Eq. 4 is estimated only once by thresholding the background alignment errors. If background, $M_t = 0$, otherwise $M_t = 1$. Layers are refined through Eq. 2. Here, we use Iterative Reweighted Least Square (IRLS) for sparse matrices. We use the first motion and layer estimates to initialize our solver. We update the motion field through optical flow [16] and refine layers again using Eq. 2. This process is repeated till convergence. We usually run Eq. 2 at most twice. Finally, we apply a temporal moving average filter to reduce flickering. For this we make sure the mean value of all frames is the same.

The solution of Eq. 2 is expressed in the form of $\|Ay - C\|$ where y is a vector of size $M = N \times v \times h$. N is the total number of frames and (v, h) are the vertical and horizontal resolutions. C is of size $80 \times v \times h$ and A is of size $80 \times v \times h \times M$. We use IRLS to solve for y. This is a large system and hence due to computational limitations we cannot solve all frames at once. Instead, we use a moving temporal window. We process the video in small chunks of n frames (see Fig. 3 (a)). For each chunk, n separations are generated and the final result is taken as the average of these outputs. Here, from every 15 frames, we selected $n = 5$ frames with an equal spacing of 3. This window is centered on every frame to process all video frames. Fig. 3 (b) compares a conditional solution against our non-conditional solution. A 'Conditional' solution is constrained by the separation of previous frames. Fig. 3 (b) shows it generates temporal flickering (see red box and line).

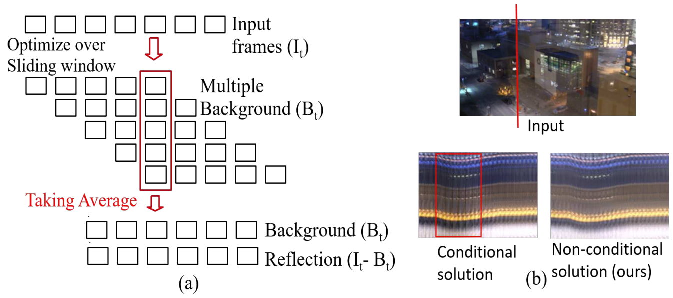

Figure 3. (a) Applying Eq. 2 for all frames at once is computationally intractable. Instead, we process windows of five frames moving with one frame. This generates five outputs for each frame. Their average is the final output. (b) A solver conditioned on previous frame separations generates more artifacts (red).

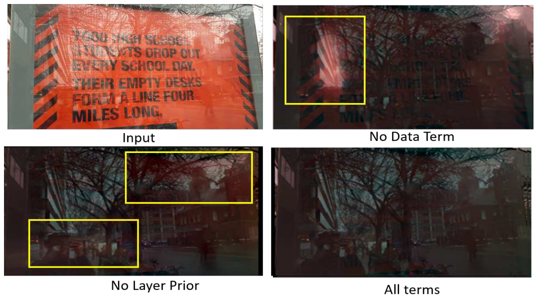

Figure 4. The impact of the different objective function terms (see yellow for artifacts). The best results are obtained with all terms.

Tab. 1 compares the main components of our approach against Xue et al. [28] and Yang et al. [29]. To our best knowledge, our method is the first method providing the *per-frame* layer separation. This is achieved by the double summation over all frames in Eq. 3. The absence of the layer prior from Yang et al. or the temporal layer redundancy from Xue et al. generates flickering, removes reflection incompletely and blurs local movements. Fig. 4 shows the impact of different objective terms. Removing any of them generates incomplete separation (see yellow boxes). Furthermore, we use iterative matching for alignment. Fig. 5 shows this generates better separation than the direct matching of Xue et al. and Yang et al.. Fig. 5 uses edge-flow for alignment.

4. Results

We have performed experiments on real data and on images generated in controlled environments. For real se-

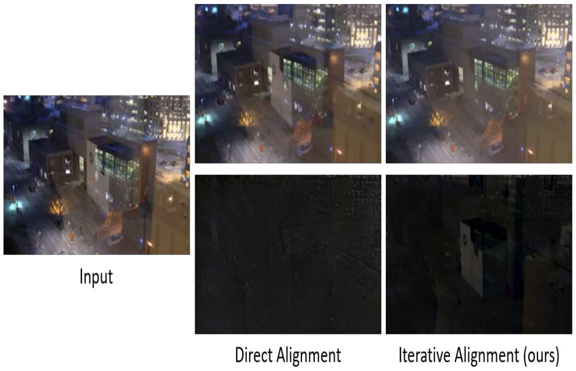

Figure 5. Direct alignment can lead to poor separation. Our iterative alignment (Eq. 6), however, generates better results.

quences we assess performance qualitatively. For controlled experiments we also assess performance quantitatively against the ground-truth. We processed 12 sequences two of which are generated under controlled settings. Advanced video extensions of the state of the art generate an incomplete separation and strong temporal inconsistencies. In addition, they blur locally moving objects. Our approach significantly improves the layer separation and the temporal coherency and better recovers local motions. Video results are on https://youtu.be/V87GGFdtDSQ.

Separation Techniques Xue et al. [28] and Yang et al. [29] are the latest separation techniques. As none of them is designed for videos, we evaluate our approach against different implementations, post-scripted '+' and '++'. The former applies the original technique using our moving window strategy of Fig. 3. We use the same window parameters, 5 frames separated by 3, with 1 frame movement. We average all separations of every frame to produce the final result. To reduce flickering, '++' uses a moving temporal average filter on '+'. In all implementations we use our iterative alignment strategy instead of the direct alignment of [28, 29]. This further improves performance (see Fig. 5). In all comparisons and for all techniques we use the same parameter values of Eq. 2 and Eq. 6. We also include results for Gai et al. [9], Li et al. [15] and Guo et al. [10].

4.1. Real Sequences

Fig. 6-11 shows the separation on different sequences with different techniques. All sequences contain strong reflections, a camera motion (angular in Fig. 9) and some contain local movements (Fig. 10 and Fig. 11). For each sequence we show two frames and zoom in on a spatio-temporal slice to show the temporal variations on one image. This visualization is the same one used in [24, 4]. Some sequences are shot outdoor (Fig. 6,8,9,11), others are shot indoor (Fig. 7) and a few shot in a mobile environment (moving bus, Fig. 10). Fig. 6 shows a sequence shot by a person moving from right to left. Xue et al. + generates poor separation and strong temporal intensity fluctuations (see yellow spatio-temporal slice). Xue et al. ++ reduces the temporal fluctuations significantly, yet still generates an incomplete removal (red boxes). Similar performance is generated for Yang et al. ++. Our approach, however, removes the ghosting artifact and generates temporally coherent results. Fig. 7 shows a sequence shot through a glass window. Here, a t-shirt and a fan are reflected in the glass window. Xue et al. + and Xue et al. ++ produces poor separation with strong temporal inconsistencies (yellow slice). Yang et al. ++ reduces such inconsistencies yet degrades the spatial removal. Our approach enhances both separation and temporal coherency. Fig. 8-9 process two sequences with Xue et al. ++ and our technique. We produce more temporally coherent results with better separation.

Fig. 10 shows a challenging sequence shot from a moving bus. The background is highly dynamic and contains a bridge, river and moving people, including a cyclist. The reflection is of the camera man. This is a common scenario where reflection removal is desirable. For this sequence we annotate reflection points every 30 frames. We propagate the labeling through SIFT-flow and outlier regions are taken as the background. We use this strategy for motion initialization for all techniques. Xue et al. ++ remove reflections, however, significantly blurs the cyclists. In addition, it blurs the river and the bridge (red spatio-temporal slice). Yang et al. ++ generate incomplete separation (green box). Our approach better removes reflection and generate temporally coherent results. We also significantly reduce Xue et al. ++'s cyclist blurring.

Fig. 11 shows an example of reflection reconstruction. Here, the camera man is shooting a glass-covered billboard. The street is reflected on the glass. The street contains building, trees and a moving person (see yellow). Xue et al. generates just one output frame and hence completely removes the moving person (yellow). Advanced video extension (Xue et al. +) recovers the person, however, still blurs him significantly. The improvement, here, is due to using our moving window strategy which accounts for some local movements. Similarly, Yang et al. + blurs the person significantly and generates a strong reddish background bleeding. Our approach outperforms all techniques and recovers the moving person with a high accuracy. Using '++' would further blur the person. Fig. 12 processes the input of Fig. 5 using Gai et al. [9], Li et al. [15] and Guo et al. [10]. Our removal in Fig. 5 outperforms all techniques.

4.2. Controlled Experiments

We processed two sequences created under controlled settings. We have the ground-truth of the underlying layers which allows us to perform subjective and objective evaluation. For objective evaluation we use two classes of metrics; the first measures the spatial similarity of the reconstructed

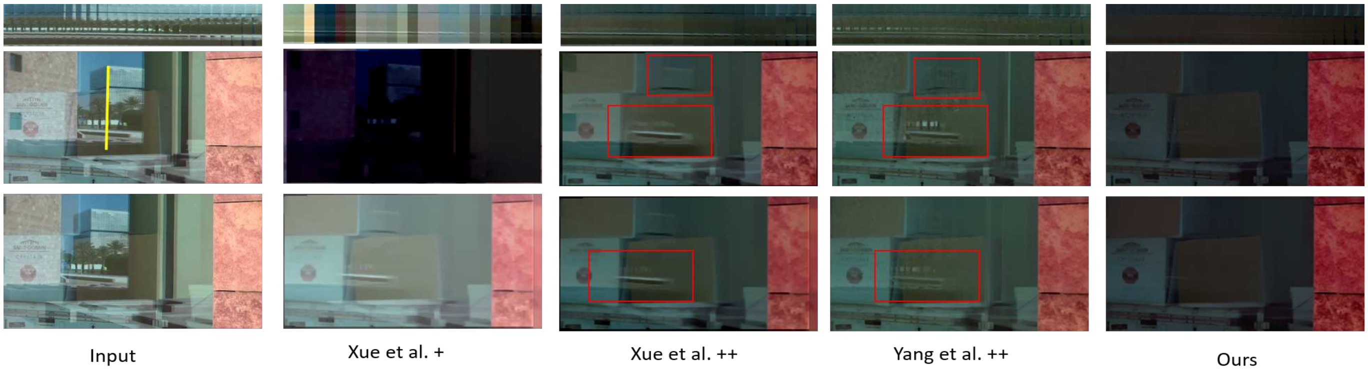

Figure 6. Reflection removal by different techniques (see Sec. 4, **Separation Techniques**, for explanation of '+' and '++'). Each column shows two frames and zoom in on the yellow spatio-temporal slice. Xue et al. + generates temporally incoherent results while Xue et al. ++ and Yang et al. ++ have ghosting artifacts (red region). Our approach improves both the removal and temporal coherency (yellow slice).

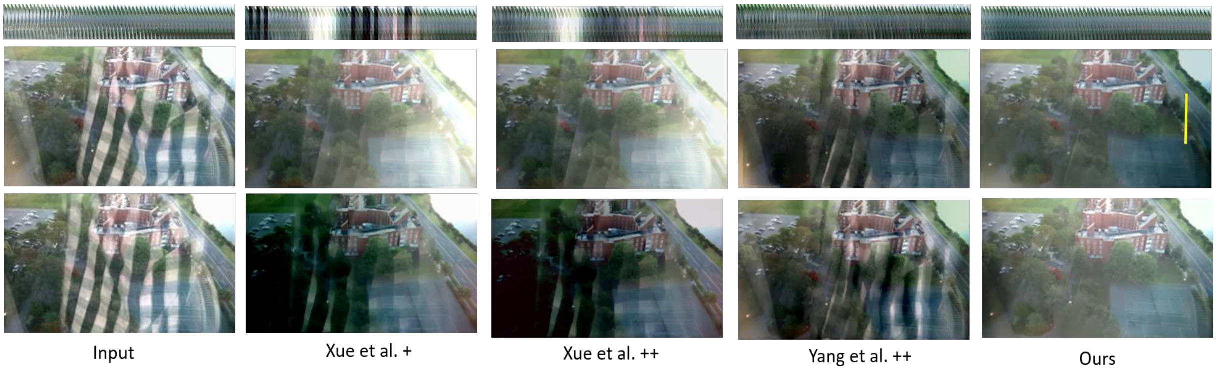

Figure 7. The reflection removal for a video. Advanced video extensions of state of the art ('+' and '++') generate poor separation with temporal flickering (yellow slice). Our approach generates significantly better separation with temporally coherent results.

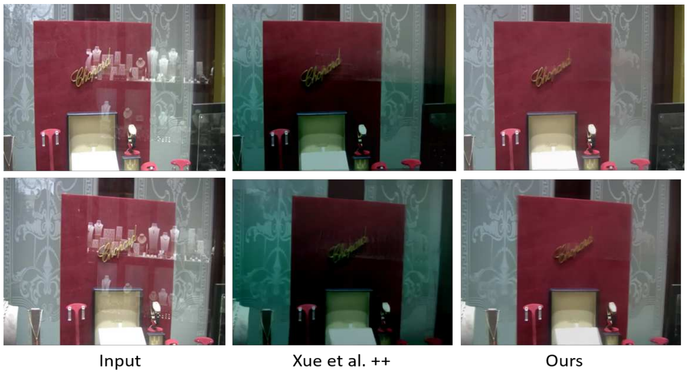

Figure 8. The reflection removal by different techniques. Our approach generates better separation than Xue et al. ++.

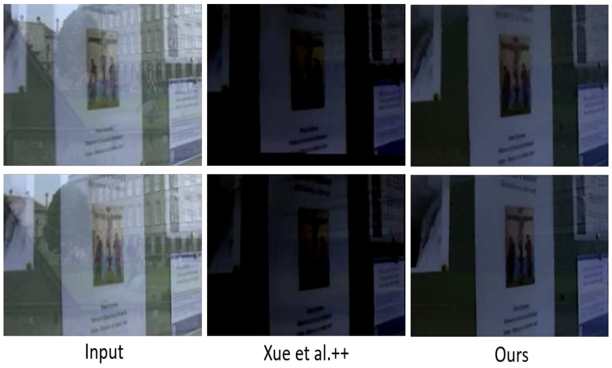

Figure 9. Reflection removal by different techniques. Our approach generates better separation than Xue et al. ++.

layers against the ground-truth while the second assess the temporal consistency. Spatial similarity is measured using Normalized Cross Correlation and the Structural Similarity Index SSIM [26]. Both metrics occupy a range between 0 and 1, where 1 means perfect separation. Both metrics account for a mean color shift to allow a more robust comparison between different techniques. We show the plot over all frames. To assess temporal coherency, we estimate the Fast Fourier Transform of the examined sequence and plot the power spectral density (PSD). We measure the summation of the high frequency components (larger than 6 Hz) and show the per frame average in the legend. The more flicker-

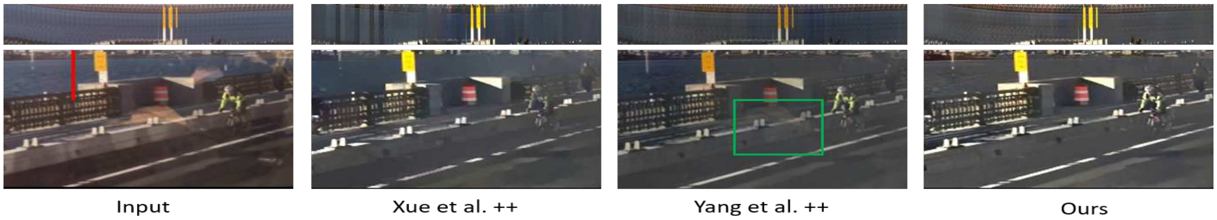

Figure 10. Our removal against advanced video extensions of the state of the art ('+' and '++'). Yang et al. ++ generates incomplete separation (see green box) while Xue et al. ++ blurs the cyclist, the sea and generate temporally incoherent results (see the red slice). Our approach significantly reduces temporal flickering, image blurriness and improves separation.

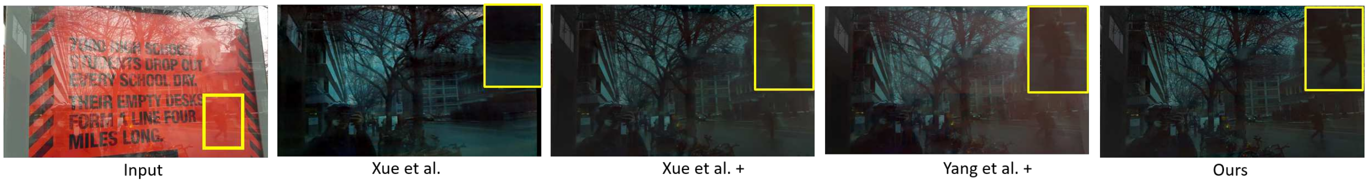

Figure 11. Reflection reconstruction example. We best reconstruct the moving person (yellow) and avoid the red bleeding of Yang et al.+.

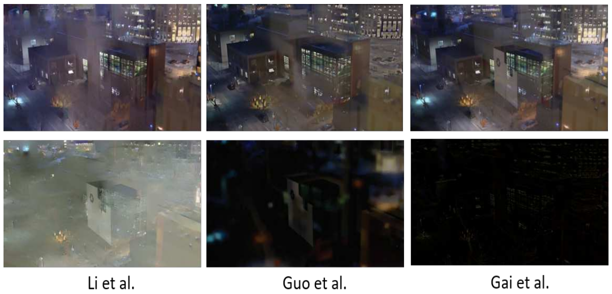

Figure 12. Processing the input of Fig. 5 with different techniques. Our result in Fig. 5 (iterative alignment) generates the best separation.

ing, the larger this quantity. Here, we are mainly interested in assessing fluctuations not generated by the original motion sources, e.g. the camera, objects and so on. Hence, we remove such motions prior to PSD calculation. The motions are controlled and therefore known.

The first controlled sequence (Fig. 13) is generated by synthetically mixing two videos through the image compositing equation of Eq. 1. We treat Lena as the foreground and a picture of buildings as the background. We synthetically add two different global movements for both layers (-1 and +1 pixels/frame in horizontal direction). We overlaid a bird on the background and gave it a third different motion (+2 pixels/frame in horizontal direction). This is to assess the ability of separation techniques in handling locally moving objects. We process the generated sequences using Xue et al. +, Xue et al. ++ and Yang et al. ++. Xue et al. + and Yang et al. ++ produce poor spatial separation while Xue et al. ++ produce a more descent one. This is captured visually and through the NCC and SSIM. In addition, all techniques have poor temporal consistency. Our approach generates a temporally coherent good separation. This is captured by the green spatio-temporal slice and the PSD. The total high frequency components of both layers for our approach is 0.0107. This compares favorably with 0.0195 for Xue et al. ++ and 0.0843 for Yang et al. ++. To assess local movements reconstruction we zoom on the bird (blue box). Our approach generates a sharper reconstruction than Xue et al. ++. This improvement is in part due to our optimization which directly estimates n background outputs at once. Xue et al. ++ estimates only one background image for all input frames and hence blur moving objects.

The second controlled sequence (Fig. 14) is generated by mixing two real objects. The background is a black box (see yellow region) and the foreground is a big red box with a small dark one (see red region). Foreground and background objects are separated by a glass to allow reflection to occur (purple). Separation techniques require the mixed sequence to be moving. In addition, the ground-truth background should undergo the exact camera motion. To achieve that we use Cinetics CineMoco Dolly and SkateTrack [3]. This allows camera motion to be defined through a controller. It also generated a smooth camera path without bumps or shakiness. We perform two rounds of shooting with the same motion settings. The first includes all foreground and background objects to generate the mixed sequence. The second is with the background object only to generate the ground-truth estimate. Our approach produces better separation over Xue et al. ++ and Yang et al. ++. This is captured both subjectively and objectively. For our approach the mean NCC and SSIM is $(0.82, 0.82)$. This compares favorably with Xue et al. ++ $(0.77, 0.78)$ and Yang et al. ++ $(0.81, 0.79)$. In addition, the total temporal high frequency components of our reconstructed background is

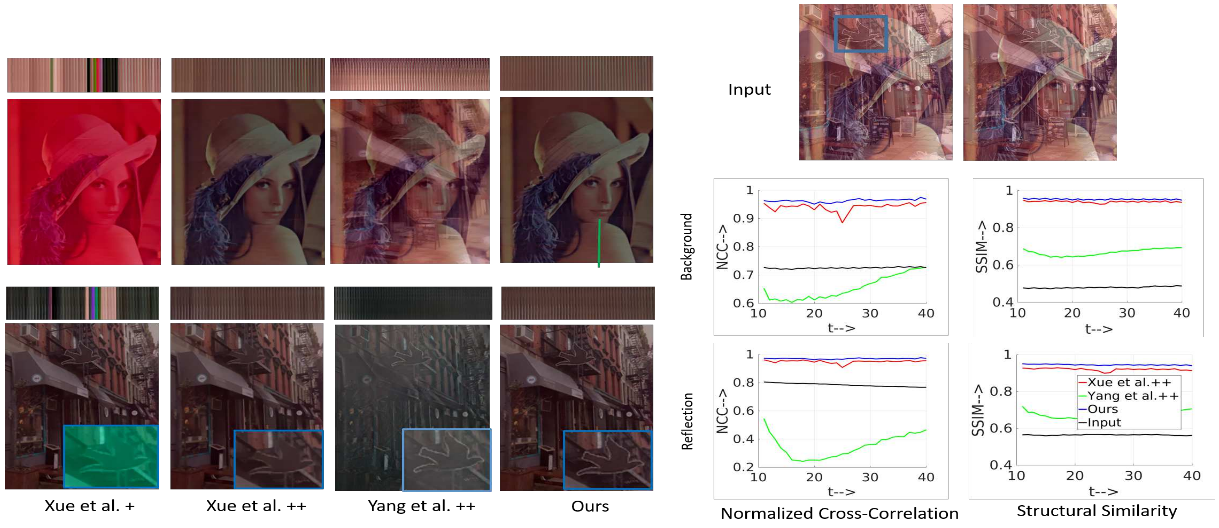

Figure 13. Evaluating different separation techniques on a controlled synthetic sequence. Here, Lenna is moving to the left while the background is moving to the right. In addition, the bird (in blue) has a third different motion. Our approach generates the best temporal coherency and separation. This is captured both subjectively (see spatio-temporal green slice) and objectively through different metrics. Our temporal PSD is 0.0107 (> 6 Hz). This compares favorably with Xue et al. ++'s 0.0195 and Yang et al. ++'s 0.0843.

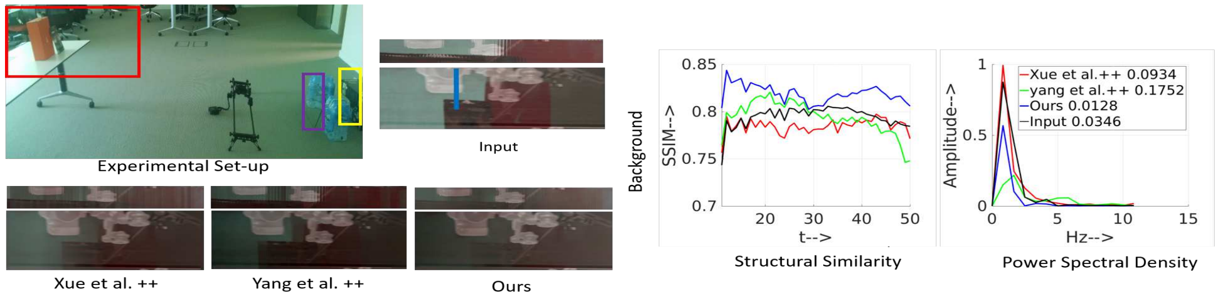

Figure 14. Processing a controlled real sequence. The background is shown in yellow and the foreground in red (experimental set-up), with a reflecting glass in between (purple). We use a camera dolly and SkateTrack to simulate and control motion. This generates the input sequence. The dolly with the same motion settings is also used to generate the ground-truth background sequence, without the foreground. Our technique generates the best separation and temporal coherency. This is observed both objectively and subjectively (blue slice). We measure the temporal PSD of the separated layers and the SSIM against ground-truth. We achieve the least PSD of 0.0128 (> 6 Hz).

0.0128. This compares favorably with 0.0934 for Xue et al. ++ and 0.01752 for Yang et al. ++.

Limitations: Our technique initially uses one homography for each layer. This assumption may not valid for scenes with strong depth variations. Our approach recovers locally moving objects (bird in Fig. 13, cyclist in Fig. 10). However, if such motion is too large in the background, it may not fit in the initial homography and can bleed to the reflection. Our technique also requires sufficient features in each layer, undergoing two different global motions. Hence, similar layers can be handled as long as sufficient features are detected. Insufficient features leads to inaccurate alignment and incomplete removal. We reduced this limitation through our iterative alignment (Fig. 5). However, it can be problematic for low contrast reflections. Finally, intensity saturated regions can bleed in the separation.

5. Conclusion

We presented the first technique for reflection removal in videos. Current removal approaches are designed for still images. Our technique takes a video as input and produces a video through a novel spatio-temporal formulation. Our objective terms aim to fully utilize spatio-temporal information. It also uses a novel iterative alignment strategy. We examined our approach on a variety of data through subjective and objective evaluations. This includes real and controlled data. We compared against advanced video extensions of the state of the art. Our results show significant reduction in temporal flickering with more complete removal. We also reduce image blur and better handle moving objects. Future work can address highly non-planar layer movements and temporally varying mixing parameters.

References

[1] A. Adams, E.-V. Talvala, S. H. Park, D. E. Jacobs, B. Ajdin, N. Gelfand, J. Dolson, D. Vaquero, J. Baek, M. Tico, H. P. A. Lensch, W. Matusik, K. Pulli, M. Horowitz, and M. Levoy. The frankencamera: An experimental platform for computational photography. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 29(4):29:1–29:12, 2010. 1

[2] A. Bronstein and M. Bronstein. Sparse ica for blind separation of transmitted and reflected images. In *International Journal of Imaging Systems and Technology*, pages 84–91, 2005. 2

[3] Cinetics. Cinemoco System. https://cinetics.com/kit/cinemoco_system/, 2014. 7

[4] M. Elgharib, M. Hefeeda, F. Durand, and B. Freeman. Video magnification in presence of large motions. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4119–4127, 2015. 5

[5] M. A. Elgharib, F. Pitie, and A. Kokaram. Reflection detection in image sequences. In *CVPR*, pages 705–712, 2011. 2

[6] M. A. Elgharib, F. Pitite, and A. Kokaram. Motion estimation for regions of reflections through layer separation. In *European Conference on Visual Media Production (CVMP)*, pages 49–58, Oct 2011. 1

[7] M. A. Elgharib, F. Pitite, A. Kokaram, and V. Saligrama. User-assisted reflection detection and feature point tracking. In *CVMP*, pages 13:1–13:10, 2013. 2

[8] H. Farid and E. Adelson. Separating reflections from images by use of independent component analysis. In *CVPR*, pages 262–267, 1999. 2

[9] K. Gai, Z. Shi, and C. Zhang. Blind separation of superimposed moving images using image statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(1):19–32, 2012. 2, 5

[10] X. Guo, X. Cao, and Y. Ma. Robust separation of reflection from multiple images. In *CVPR*, pages 2195–2202, 2014. 1, 2, 5

[11] J. Kopf, F. Langguth, D. Scharstein, R. Szeliski, and M. Goesele. Image-based rendering in the gradient domain. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 32(6):199:1–199:9, 2013. 2

[12] M. Lang, O. Wang, T. Aydin, A. Smolic, and M. Gross. Practical temporal consistency for image-based graphics applications. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 31(4):34:1–34:8, 2012. 2

[13] A. Levin and Y. Weiss. User assisted separation of reflections from a single image using a sparsity prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29:1647–1654, 2007. 1, 2

[14] A. Levin, A. Zomet, and Y. Weiss. Separating reflections from a single image using local features. In *CVPR*, pages 306–313, 2004. 1, 2

[15] Y. Li and M. S. Brown. Exploiting reflection change for automatic reflection removal. In *ICCV*, pages 2432–2439, 2013. 2, 5

[16] C. Liu. Beyond pixels: Exploring new representations and applications for motion analysis. *MIT, PhD Thesis*, 2009. 4

[17] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(5):978–994, 2011. 2, 4

[18] B. Sarel and M. Irani. Separating transparent layers through layer information exchange. In *ECCV*, pages 328–341, 2004. 1, 2

[19] B. Sarel and M. Irani. Separating transparent layers of repetitive dynamic behaviors. In *ICCV*, pages 26–32, Oct 2005. 1, 2

[20] Y. Shih, D. Krishnan, F. Durand, and W. T. Freeman. Reflection removal using ghosting cues. In *CVPR*, pages 3193–3201, June 2015. 1, 2

[21] S. Sinha, J. Kopf, M. Goesele, D. Scharstein, and R. Szeliski. Image-based rendering for scenes with reflections. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 31(4):100:1–100:10, 2012. 2

[22] R. Szeliski, S. Avidan, and P. Anandan. Layer extraction from multiple images containing reflections and transparency. In *CVPR*, pages 246–253, 2000. 1, 2

[23] Y. Tsin, S. B. Kang, and R. Szeliski. Stereo matching with reflections and translucency. In *CVPR*, pages 702–709, 2003. 2

[24] N. Wadhwa, D. Rubinstein, Fredo, and W. T. Freeman. Phase-based video motion processing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 32(4), 2013. 5

[25] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action Recognition by Dense Trajectories. In *CVPR*, pages 3169–3176, 2011. 3

[26] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)*, 13(4):600–612, April 2004. 6

[27] Y. Weiss. Deriving intrinsic images from image sequences. In *ICCV*, pages 68–75, 2001. 1, 2

[28] T. Xue, M. Rubinstein, C. Liu, and W. T. Freeman. A computational approach for obstruction-free photography. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 34(4):79:1–79:11, 2015. 1, 2, 3, 4, 5

[29] J. Yang, H. Li, Y. Dai, and R. T. Tan. Robust optical flow estimation of double-layer images under transparency or reflection. In *CVPR*, 2016. 1, 2, 4, 5