Out-of-boundary View Synthesis Towards Full-Frame Video Stabilization

Yufei Xu[1] Jing Zhang[1] Dacheng Tao[1,2]
[1]The University of Sydney, Australia
[2]JD Explore Academy, China
yuxu7116@uni.sydney.edu.au jing.zhang1@sydney.edu.au dacheng.tao@gmail.com

Abstract

*Warping-based video stabilizers smooth camera trajectory by constraining each pixel's displacement and warp stabilized frames from unstable ones accordingly. However, since the view outside the boundary is not available during warping, the resulting holes around the boundary of the stabilized frame must be discarded (i.e., cropping) to maintain visual consistency, and thus does leads to a tradeoff between stability and cropping ratio. In this paper, we make a first attempt to address this issue by proposing a new **O**ut-of-boundary **V**iew **S**ynthesis (OVS) method. By the nature of spatial coherence between adjacent frames and within each frame, OVS extrapolates the out-of-boundary view by aligning adjacent frames to each reference one. Technically, it first calculates the optical flow and propagates it to the outer boundary region according to the affinity, and then warps pixels accordingly. OVS can be integrated into existing warping-based stabilizers as a plug-and-play module to significantly improve the cropping ratio of the stabilized results. In addition, stability is improved because the jitter amplification effect caused by cropping and resizing is reduced. Experimental results on the NUS benchmark show that OVS can improve the performance of five representative state-of-the-art methods in terms of objective metrics and subjective visual quality.*[1]

1. Introduction

With the increased demand for high-quality video using handheld devices, video stabilization has become increasingly important, as such videos always contain undesirable jitter. Many video stabilization methods have been proposed to eliminate jitter in unstable videos for a better visual experience [31, 5, 38, 36, 37], and can facilitate many other computer vision tasks [3, 27, 42, 4, 41, 40].

Warping-based stabilizers [24, 21, 31, 36, 33, 37, 38] perform stabilization by first estimating and then smoothing the camera trajectories. The stabilized video is warped from the unstable video based on the pixel displacement field obtained from the transformation between the shake and smoothed trajectories. Unfortunately, some of the requisite source pixels during warping lie outside the boundary of the current unstable frame, inevitably leading to holes near the boundary of the stabilization result. To maintain visual consistency, cropping and resizing operations are employed to discard these holes, but may result in a reduction of the effective frame size, a change in the frame aspect ratio, and an amplification of the jitter. Previous approaches mitigate this problem by reducing the area of these out-of-boundary pixels, *i.e.*, by limiting the maximum deformation displacement [18, 19, 8, 7]. This constraint makes stability and crop ratio a compromise. It provides smoother trajectories with large cropping area and vice versa, both of which are not ideal for a better visual experience. Is it possible to maintain stability while increasing the cropping ratio to get (near) full frame stabilized results?

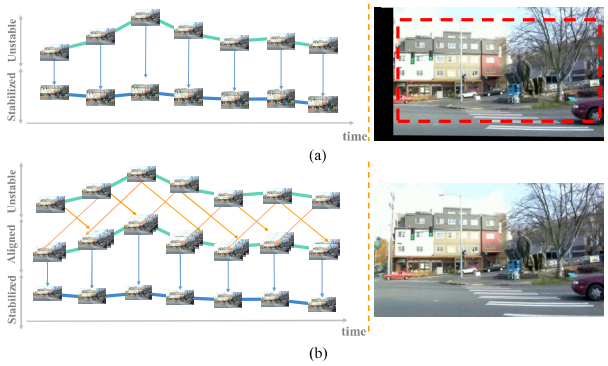

Figure 1. (a) Previous warping-based video stabilization methods only use the current frame to get the stabilized frame by warping. Since the view outside the boundary is not available during warping, the resulting holes around the boundary of the stabilized frame must be cropped. (b) Our OVS method leverages neighboring frames to expand the area of the out-of-boundary view of each current frame. Therefore the subsequent warping process can find requisite candidate pixels for obtaining a full-frame stable frame.

[1]The code is publicly available at code.

The interpolation-based stabilizers [5] provide a solution to achieve this goal by iteratively interpolating intermediate frames from adjacent frames, including those pixels that lie in the out-of-boundary view of the current frame. These methods implicitly exploit the property that the content in adjacent frames and within each frame follows spatial coherence, a property that has been widely adopted in SFM [9, 32], video inpainting [15, 12, 39], and super-resolution [16, 17, 44]. However, this property has not been explored in previous warping-based methods for filling the out-of-boundary pixels. The most intuitive way to exploit this property is to use interpolation to fill the black holes after warping. Unfortunately, some valuable content may have been permanently discarded because it was not sampled during the warping process when the stable frames were obtained, making this post-warping inpainting method invalid here as shown in [5]. In contrast, we try to investigate pre-warping extrapolation to alleviate this problem, which aims to synthesize the out-of-boundary view of each frame by exploiting the property of spatial coherence, thus facilitating the subsequent warping process to sample enough pixels as needed.

Specifically, we propose a new **O**ut-of-boundary **V**iew **S**ynthesis (OVS) method in this paper, which consists of two stages, in which the view outside the boundaries is inferred from the adjacent frames by aligning them to each reference frame. In the first coarse alignment stage, the adjacent frames are roughly aligned with the reference frame using a grid-based motion estimate. Afterwards, a second fine alignment stage is introduced to handle subtle misalignment and refine results. It first calculates the optical flow, then predicts the optical flow in the out-of-boundary views via affinity propagation, and finally warps pixels from adjacent frames according to the predicted optical flows. This process is iteratively carried out to gradually align distant neighboring frames to the current frame to expand the area of the out-of-boundary view, so that the subsequent warping process can find needed candidate pixels to obtain a stable frame. Experimental results on the NUS benchmark show that OVS can be plugged in five representative warping-based methods, significantly improving the cropping ratio of stabilized results.

In summary, the contribution of this work is threefold:

1) We make a first attempt to improve warping-based video stabilizers towards full-frame stabilization by extrapolating the requisite out-of-boundary view during warping.

2) We propose a two-stage coarse-to-fine method for out-of-boundary view synthesis by exploiting the spatial coherence in the video.

3) Experimental results on publicly available datasets show that the proposed method can serve as a plug-and-play module to significantly improve both grid-based and pixel-based warping methods.

2. Related Work

2.1. Warping-based Video Stabilization

A representative solution for video stabilization is to estimate the warping field from unstable frames to stabilized ones. Traditional methods typically follow a three-step procedure, first estimating the trajectory, then smoothing the trajectory, and finally obtaining the stabilized frame from the unstable one based on the warping field. The warping field is generated by computing the transformation between the original trajectory and the smoothed trajectory. These methods can constrain the maximum transformation to reduce the area of the out-of-boundary view needed during the warping process, yielding a high cropping ratio while leading to low stability. For example, Subspace [19] minimizes the displacement of pixels after smoothing the trajectory, which is fitted by a polynomial curve. A similar strategy is utilized in CPW [18, 8]. Bundled [23], SteadyFlow [24], and MeshFlow [21] try to minimize the transformation as well as the motion between adjacent frames to reduce the area of the out-of-boundary view. Liu *et al*. [22] uses a depth camera for video stabilization and limits the maximum rotation and translation transformations. L1Stabilizer [8] limits the range of each element in the warping matrix to reduce the area of the out-of-boundary view. Deep learning-based stabilizers learn to regress the unstable-to-stable warping field for stabilization or stabilized frames directly. By leveraging the ground truth stable frames as supervisory signal [31, 33], the warping field is implicitly constrained to produce stabilized results. Since some pixels on the stabilized frame are not available due to the absence of pixel correspondence during warping, their losses are not calculated, implying that cropping is still needed to get the desired final result.

These methods only make a tradeoff between cropping ratio and stability as the need of out-of-boundary pixels during warping is reduced rather than satisfied. In contrast, we propose a new method, named OVS, that explicitly extrapolates the requisite out-of-boundary pixels for warping, helping warping-based stabilizers to achieve full-frame video stabilization. It offers a new perceptive on video stabilization and does not require the transformation to be limited to a small range. In addition, OVS can serve as a plug-and-play module to significantly improve both grid-based and pixel-based warping methods.

2.2. Interpolation-based Video Stabilization

Towards cropping-free video stabilization, DIFRINT [5] proposes to treat stabilization as a frame generation problem, where the stabilized frames are intermediate frames generated by interpolating from adjacent frames in an iterative manner. However, due to the large jitter in unstable videos, the estimated optical flow for interpolation is not

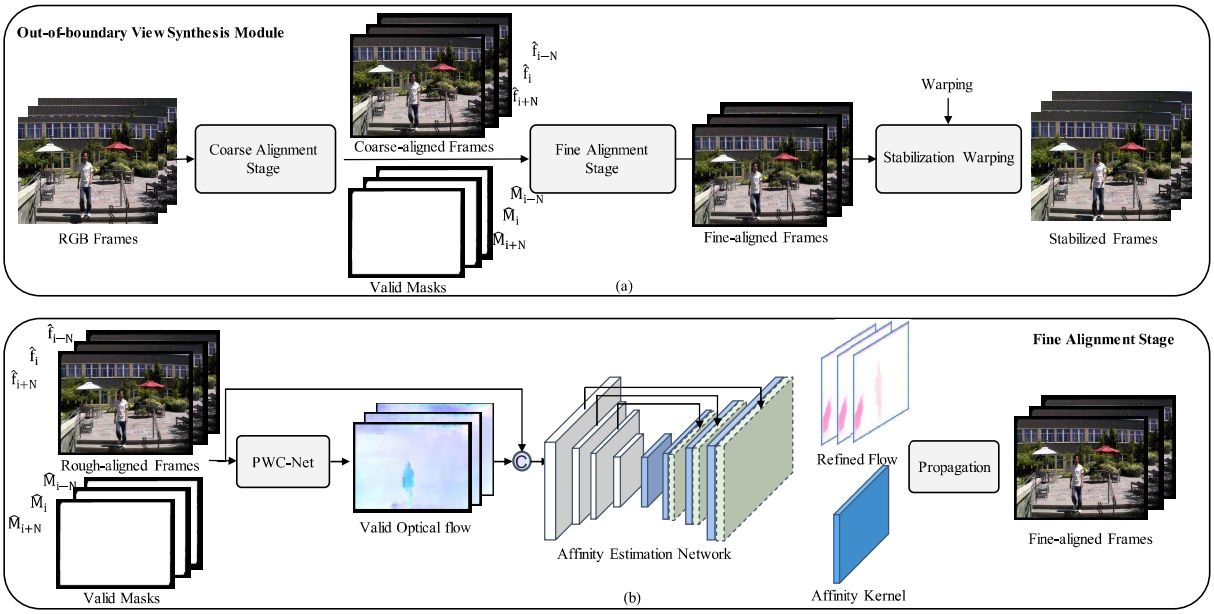

Figure 2. (a) OVS consists of a two alignment stages that align adjacent frames to each reference frame from coarse to fine. (b) The fine alignment stage first estimates optical flow between the roughly aligned frame and the reference frame inside the boundary, then propagates them outside the boundary based on affinity, and finally the out-of-boundary pixels can be extrapolated based on the optical flow.

reliable, leading to distortions or ghost artifacts, especially around the boundaries of dynamic objects. Nevertheless, it is insightful to exploit the spatial coherence of adjacent frames for full-frame video stabilization. In this paper, we also investigate the impact of spatial coherence, but from a different perspective, *i.e.*, extrapolating the out-of-boundary view for benefiting warping-based methods rather than interpolation for video stabilization directly. Together with warping-based stabilizers, they can obtain better stabilized results with less distortions or ghost artifacts while maintaining a high cropping ratio as well as good stability.

2.3. Image Alignment

Obtaining out-of-boundary view of each frame can be obtained by aligning adjacent frames to the reference one. Traditional image alignment methods always detect feature points first, *e.g.*, SIFT [25], SURF [2], ORB [29], LIFT [35], then select robust feature points from them, *e.g.*, by RANSAC [9], and finally use them to calculate the transformation for alignment. Recently, several deep learning-based image alignment methods are proposed [6, 26, 43] by either supervised learning or unsupervised learning. For example, Zhang *et al*. [43] estimates a global homography between adjacent frames and uses it for alignment. It is noteworthy that using a single global homography may be ineffective when there is a large movement of the camera pose, which is very common in unstable videos in the video stabilization task. Nevertheless, because of its superiority over traditional methods, we use it as the baseline method for out-of-boundary view synthesis. In contrast, we propose a two-stage coarse-to-fine method, which can deal with large camera movements and dynamic objects more effectively.

3. Methods

The proposed OVS method aims to synthesize the out-of-boundary view for video stabilization. It consists of a coarse alignment stage and a fine alignment stage. As shown in Figure 2, the coarse alignment stage takes each current frame together with its neighboring frames as input. It aligns each neighboring frame to the reference one and generates a mask to indicate the valid out-of-boundary view after alignment. The second fine alignment stage takes the aligned frames and their masks as input for further refinement. These two stages are carried out alternatively to gradually align distant frames to the reference one and expand the area of out-of-boundary view. The details of the two stages will be discussed as follows.

3.1. Coarse Alignment

We take the motion estimation module in DUT [34] for motion estimation in the coarse alignment stage. It first detects keypoints and estimates their motion in each frame, then propagates the motion of keypoints to a set of predefined grids, and finally estimates the homography for each grid. The keypoints are robust to noise and illumination variation while the grid-based estimation can handle dynamic objects and large jitter. In addition, a multi-planar

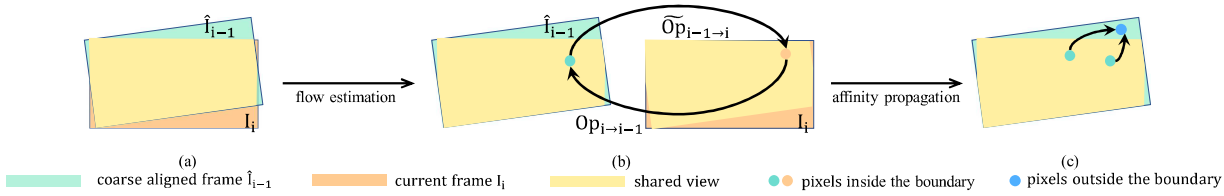

Figure 3. Illustration of the fine alignment stage. (a) The roughly aligned frames from the first stage. (b) Estimation of the optical flow $Op_{i\to i-1}$ from the reference frame I_i to the aligned frame $\widehat{I}_{i-1}$ and the reverse flow $\widetilde{Op}_{i-1\to i}$ from $\widehat{I}_{i-1}$ to I_i. (c) The optical flows inside the boundary are propagated outside the boundary using affinity. The out-of-boundary pixels can be extrapolated based on the optical flow.

estimation strategy is used to deal with the complex motion patterns in real scenes. After obtaining the motion of the mesh vertices, the mesh-based homography matrices are computed and used for coarse alignment. The aligned frames are used as the input for the fine alignment stage. Specifically, the frames in the input unstable video are assumed to be $\{I_i \mid i = 1, \cdots, n\}$, where n is the total number of frames and I_i represents the ith frame. The size of the input frames is $H_o \times W_o$, where H_o is 480 and W_o is 640. We employ zero-padding outside the boundary of each frame before warping, *i.e.* 80 pixels in each direction and using masks $\{M_i \mid i = 1, \cdots, n\}$ to denote the valid regions after padding. The final output of the coarse alignment module is $\{\hat{I}_{i\to i+1}, \hat{M}_{i\to i+1} \mid i = 1, \cdots, n\} \cup \{\hat{I}_{i\to i-1}, \hat{M}_{i\to i-1} \mid i = 1, \cdots, n\}$, where the former set is generated from a forward pass and the latter set is generated from a backward pass. Note that we keep $\hat{I}_{n\to n+1} \triangleq I_n$, $\hat{M}_{n\to n+1} \triangleq M_n$, $\hat{I}_{1\to 0} \triangleq I_1$, and $\hat{M}_{1\to 0} \triangleq M_1$. The coarse-aligned frames ($\hat{I}$) and corresponding masks ($\hat{M}$) are of size $H_m \times W_m$, where $H_m = 640$ and $W_m = 800$.

3.2. Fine Alignment

The coarse aligned frames can serve as good initial out-of-boundary view synthesis results, although there is some subtle misalignment due to the influence of dynamic objects or large jitter. To further refine the results, we propose a fine alignment (FA) stage. In the following, we denote $\hat{I}_{i\to i+1}$ as $\hat{I}_i$ and $\hat{M}_{i\to i+1}$ as $\hat{M}_i$ and only present the forward pass for simplicity. As shown in Figure 2(b), the fine alignment stage first estimates optical flow between the roughly aligned frame $\widehat{I}_{i-1}$ and the reference frame I_i inside the boundary, then propagates them outside the boundary based on affinity, and finally the out-of-boundary pixels can be extrapolated based on the optical flow from the reference frame as the refined result.

Taking $\{\widehat{I}_{i-1}, I_i\}$ as input, the optical flow $Op_{i\to i-1}$ from I_i to $\widehat{I}_{i-1}$ is estimated using PWCNet [30]:

$$Op_{i\to i-1} = PWCNet(I_i, \widehat{I}_{i-1}) \odot M_i, \tag{1}$$

where $\odot$ denotes element-wise multiplication. Note that PWCNet has been widely used for optical flow estimation in many stabilizers [5, 34] due to its good performance. Then, a flow reverse (FR) layer [1, 11] is employed to get the reversed optical flow from $\widehat{I}_{i-1}$ to I_i:

$$(\widetilde{Op}_{i-1\to i}, \widetilde{M}_i) = FR(Op_{i\to i-1}, M_i). \tag{2}$$

The reason that we use flow reverse to estimate $Op_{f-1\to i}$ instead of directly estimating it from $\widehat{I}_{i-1}$ to I_i using PWC-Net is that some pixels in $\widehat{I}_{i-1}$ may correspond to the pixels outside the boundary of I_i, therefore leading to erroneous optical flow. Moreover, the flow reverse layer can output a mask $\widetilde{M}_i$ which indicates whether or not a pixel in $\widehat{I}_{i-1}$ has a corresponding pixel in I_i, *i.e.*, the yellow shared view of $\widehat{I}_{i-1}$ and I_i illustrated in Figure 3.

Given the $Op_{i-1\to i}$ inside the shared view, we want to get the optical flow outside the boundary such that the pixels there can be extrapolated from I_i accordingly. Following the spatial coherence property within the frame, we argue that the motion of static objects should be coherent locally. Based on this assumption, we propose to estimate the affinity kernels from the color and structure information and use them to propagate the optical flow inside the shared view to the outside of the boundary as illustrated in Figure 3(c).

Technically, we devise an encoder-decoder network as shown in Figure 2(b) to estimate the affinity kernels from I_i, $\widehat{I}_{i-1}$, M_i, $\widehat{M}_{i-1}$, $\widetilde{M}_i$, $\widetilde{Op}_{i-1\to i}$, G_i, and $\widehat{G}_{i-1}$, where G_i ($\widehat{G}_{i-1}$) denotes the edge map extracted from I_i ($\widehat{I}_{i-1}$) using the Sobel filter, which encodes the structure information. We use ResNet-50 [10] as the backbone encoder. The decoder takes the features from the last layer of the encoder as input and gradually upsamples the features to decode the features to the original resolution. To fully utilize both high- and low-level features, we follow the UNet-like structure [28] to concatenate the feature from the encoder and previous layer of the decoder as the input of the next decoder layer. Each decoder layer is composed of three convolution layers with batch normalization. After getting the output of the decoder, a convolution layer (denoting as $Affinity(\cdot)$) is employed to predict pixel-wise affinity kernels. It also predicts a refined flow to use in the subsequent propagation process by employing a separate convolution

layer (denoting as *RefinedFlow*$(\cdot)$), *i.e.*,

$$F_i^{\{c\}} = \textit{Encoder}([I_i, \widehat{I}_{i-1}, M_i, \widehat{M}_{i-1}, \widetilde{M}_i, \widetilde{Op}_{i-1\to i}, Op_{i-1\to i}, G_i, \widehat{G}_{i-1}]), c \in \{1,2,3,4\}, \tag{3}$$

$$D_i^c = \textit{Decoder}([D_i^{c-1}, F_i^{4-c}]), c \in \{1,2,3,4\}, \tag{4}$$

$$\begin{aligned} \kappa_i &= \textit{Affinity}(D_i^4) \\ B_i &= \textit{RefinedFlow}(D_i^4). \end{aligned} \tag{5}$$

The affinity matrix κ_i is of size $[H_m, W_m, (2r+1)^2]$, where r is the radius of affinity kernel, *i.e.*, 4 in this paper. B_i is the refined flow of size $[H_m, W_m, C]$, *i.e.*, which also provides initial estimate for the out-of-boundary view. Then, we use the affinity matrix κ_i, the refined optical flow B_i, and the mask $\widetilde{M}_i$ to propagate the optical flow from pixels in $\widetilde{M}_i$ to the out-of-boundary pixels. Mathematically, this process can be formulated as follows:

$$\begin{aligned} B_i^{t+1}[u,v] = \hat{\kappa}_i[u,v,r] \cdot B_i^0[u,v] + \\ \sum_{a,b=-r,a,b\neq 0}^{r} \hat{\kappa}_i[u,v,ar+b] \cdot B_i^t[u-a, v-b], \end{aligned} \tag{6}$$

$$\hat{\kappa}_i[u,v,ar+b] = \frac{\kappa_i[u,v,ar+b]}{\sum_{a,b\neq 0} \kappa_i[u,v,ar+b]}, \tag{7}$$

$$\hat{\kappa}_i[u,v,r] = 1 - \sum_{a,b\neq 0} \hat{\kappa}_i[u,v,ar+b], \tag{8}$$

where u and v denote the 2D location of each pixel, t denotes the number of iterations, and $B_i^0 \triangleq B_i$.

3.3. Convergence Analysis

We provide a brief theoretical analysis here to show that the propagation will not cause the propagated flow to explode as in [20, 4]. For clarity, we denote $\lambda_{u,v} = \sum_{a,b\neq 0} \hat{\kappa}_i[u,v,ar+b]$, vectorize B_i to the shape of $[H_m \times W_m, c]$, and use the same symbol without causing ambiguity. Eq. (6) can be rewritten as:

$$\begin{aligned} B_i^{t+1} &= \begin{bmatrix} 0 & \hat{\kappa}_i[0,0,1*r+0] & \cdots & 0 \\ \hat{\kappa}_i[1,0,-1*r+0] & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \cdots & 0 \end{bmatrix} \cdot B_i^t \\ &+ \begin{bmatrix} 1-\lambda_{0,0} & 0 & \cdots & 0 \\ 0 & 1-\lambda_{1,0} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \cdots & 1-\lambda_{H_m,W_m} \end{bmatrix} \cdot B_i^0 \\ &\triangleq AB_i^t + (I-D)B_i^0, \end{aligned} \tag{9}$$

where A and D denote the former and latter matrices, respectively. The partial differential of Eq (9) w.r.t. t is:

$$\partial_t B_i^{t+1} = A \partial_t B_i^t. \tag{10}$$

Therefore, we have:

$$||\frac{\partial_t B_i^{t+1}}{\partial_t B_i^t}|| = ||A|| \leq \lambda_{max} = max \sum_{a,b\neq 0} \hat{\kappa}_i[u,v,ar+b]. \tag{11}$$

Note that, the maximum gradient will always be less than 1 because we use the normalized kernel value $\hat{\kappa}$ during the propagation process. Thus the stability of the flow propagation based on affinity can be guaranteed. Besides, since the refined flow inside the shared view is much more reliable, we use a slow update strategy to preserve their flow values during the propagation process. To be specific, we use a ratio of 0.9 for updating as follows:

$$B_i^t = (1 - 0.9 \times \widetilde{M}_i) B_i^t + 0.9 \times B_i^0 \widetilde{M}_i. \tag{12}$$

3.4. Training Objective

The loss function of the coarse alignment stage are the same as the DUT [34]. For the fine alignment stage, we use robust L1 loss during training. Given the propagated optical flow B_i^t, we can extrapolate the pixel value outside the boundary in $\widehat{G}_{i-1}$, $\widehat{I}_{i-1}$, and $\widehat{M}_{i-1}$, respectively, *i.e.*

$$\widehat{G}_{i-1}^e = \textit{Extrapolate}(\widehat{G}_{i-1}, B_i^t). \tag{13}$$

$$\widehat{I}_{i-1}^e = \textit{Extrapolate}(\widehat{I}_{i-1}, B_i^t). \tag{14}$$

$$\widehat{M}_{i-1}^e = \textit{Extrapolate}(\widehat{M}_{i-1}, B_i^t). \tag{15}$$

Then, we can calculate the L1 loss between the above predictions and their ground truth (I_i^g, G_i^g), *i.e.*,

$$L_I = |\widehat{I}_{i-1}^e - I_i^g + \epsilon| \odot \hat{M}_{i-1}^e, \tag{16}$$

$$L_G = |\widehat{G}_{i-1}^e - G_i^g + \epsilon| \odot \hat{M}_{i-1}^e, \tag{17}$$

where ϵ is a small value for stability, *i.e.*, 10^{-12} in this paper. Note that there may be a trivial propagation solution, where very element in the extrapolated mask $\widehat{M}_{i-1}^e$ based on B_i^t is zero. To address this issue, we add a regularization loss on $\widehat{M}_{i-1}^s$ as follows:

$$L_M = \textit{MSE}(\widehat{M}_{i-1}\widehat{M}_{i-1}^e, \widehat{M}_{i-1}), \tag{18}$$

which penalizes the shrinkage of the mask region after propagation. The final training objective function for the fine alignment stage is $L = L_I + 2 \times L_G + 2 \times L_M$, where the loss weights are set empirically.

Note that although we adopt supervised training for OVS, it does not require paired data. In our experiments, we prepare the training data by cropping the unstable frames. Specifically, we first randomly crop a region of size 800×640 from each unstable frame of size $1280 * 720$ and use it as the ground truth I_i^g. Then, we crop its central part of size 640×480 as I_i. The left surrounding part in I_i^g is indeed the out-of-boundary view w.r.t. I_i.

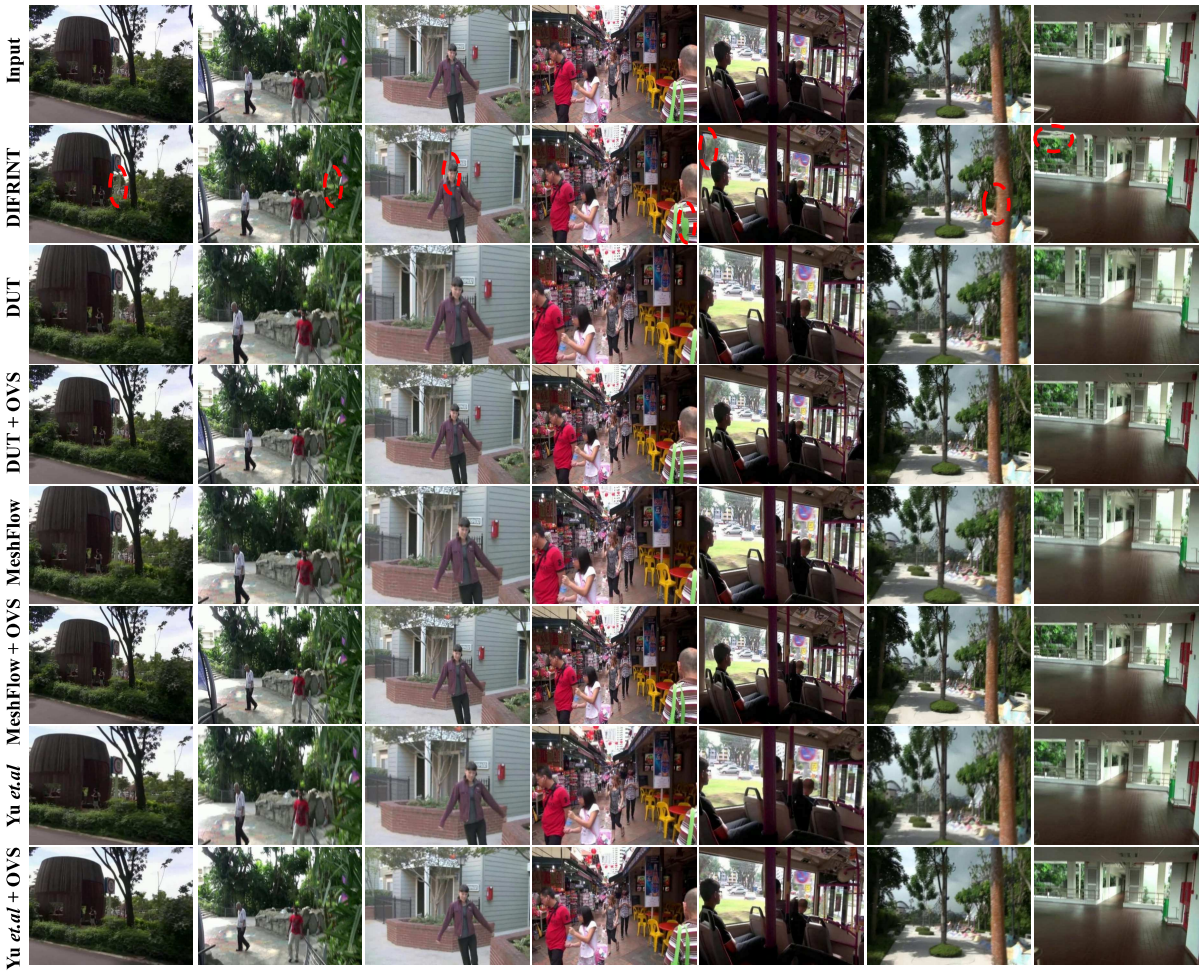

Figure 4. Subjective comparison of DIFRINT [5], DUT [34], MeshFlow [21] and Yu *et al.* [38]. "+OVS" denotes the results from corresponding stabilizer integrated with our OVS. Red circles highlight the ghost artifacts in DIFRINT's results.

4. Experiments

4.1. Implementation Details

We only use the unstable videos in the DeepStab [31] dataset for training and validation. Specifically, fifty videos are used for training and other ten videos are used for validation. We use the Adam [13] optimizer with beta (0.9, 0.99) during training. The learning rate is set to 2e-4 and decays by 0.5 every 30 epochs. We train the coarse alignment stage for 50 epochs and the fine alignment stage for 200 epochs, respectively. The training process takes about 36 hours on a single NVIDIA V100 GPU. We test the performance of our OVS model on the NUS [23] dataset.

4.2. Quantitative Evaluation

To demonstrate the versatility and effectiveness of our proposed OVS, we select several representative warping-based stabilizers and integrated OVS into these stabilizers as a plug-and-play module. Specifically, PWStabNet [45] and Yu *et al.* [38] utilize pixel-based waring for stabiliza-

Table 1. Average metrics on the NUS [23] dataset. * means the DUT stabilizer integrated with our full-frame version OVS.

	Cropping	Distortion	Stability
DIFRINT	0.949	0.854	0.823
PWStabNet	0.877	0.924	0.830
PWStabNet + **OVS**	0.959	**0.957**	0.832
Yu *et al.*	0.827	0.722	0.814
Yu *et al.* + **OVS**	0.922	0.784	0.834
StabNet	0.676	0.731	0.741
StabNet + **OVS**	0.763	0.829	0.748
MeshFlow	0.770	0.673	0.813
MeshFlow + **OVS**	0.898	0.683	0.823
DUT	0.867	0.895	0.845
DUT + **OVS**	<u>0.967</u>	0.926	<u>0.847</u>
DUT + **OVS***	**0.999**	<u>0.944</u>	**0.849**

tion while StabNet [31], Meshflow [21], and DUT [34] are grid-based warping stabilizers. OVS is integrated with these stabilizers by replacing the warping steps and keeping the rest of each method unchanged.

The results are summarized in Table 1. We use three metrics following Bundled [23] for performance evaluation, including cropping ratio, distortion, and stability, which are better for larger values. DIFRINT is the interpolation-based method, which performs better in terms of cropping ratio while falls behind warping-based methods like PWStabNet and DUT in terms of stability and distortion. It is noteworthy that the cropping ratio of DIFRINT is smaller than 1 although it does not require cropping. We suspect that this is because the stabilizer learned a small zoom in effect during the iterative interpolation process, as shown in the Figure 4. More analysis is provided in the *supplementary material*.

In addition, it can be seen that OVS can significantly improve the cropping ratio of all the warping-based methods. Meanwhile, the distortion and stability of these methods also increase as a by-product since larger cropping ratio means less cropping and resizing of the stabilized frames, therefore reducing the jitter amplification effect. Beside, since cropping aims to remove the black holes around the boundary, it does not guarantee to keep the original frame aspect ratio. Consequently, when the holes dominate in one direction, there may be large distortions in the result, *i.e.*, a lower distortion metric. In contrast, after extrapolating the out-of-boundary view via the proposed OVS, the frame aspect ratio can be better preserved, resulting in larger distortion metrics. We also notice that when applying our OVS in DUT, the cropping ratio is 0.967, which is close to 1. Most of the holes lie in texture-less regions like sky. We further use a simple trick to fill in the holes using nearest neighbor interpolation during warping. This trick leads to full-frame stabilization results as shown in the last row in Table 1.

4.3. Qualitative Evaluation

We also provide some subjective results of DUT [34], Meshflow [21], and Yu *et al.* [38] with and without our OVS as well as DIFRINT [5] for comparison in Figure 4. It can be seen that without the OVS, there is a significant content losses in the DUT, Meshflow, and Yu *et al.*'s results due to the absence of out-of-boundary views during warping. This content loss can be mitigated by extrapolating the out-of-boundary view using our OVS as shown in the fourth, sixth, and last rows. The results of DIFRINT suffer from large distortions, as highlighted by the red circle, especially when there are severe jitters between neighboring frames and dynamic objects. It is because the optical flow extracted for interpolation is not accurate due to large jitters and dynamic objects, leading to ghost artifacts around object boundaries. Besides, the results in DIFRINT suffer from small content loss, *e.g.*, the bottom left flag in the second column and the zoom-in effect in the sixth column. In contrast, OVS mitigates these issues, as shown in the results of DUT+OVS. Based on both the quantitative and qualitative evaluation results, OVS demonstrates its effectiveness in helping warping-based stabilizers to better video stabilization results with a high cropping ratio, *e.g.*, DUT+OVS achieves near full-frame stabilization results.

4.4. Ablation Study

4.4.1 Influence of Coarse and Fine Alignment

Figure 5. Results of DUT with a SOTA image alignment method [43] and our OVS. Red arrows indicate the misalignment.

Table 2. Ablation study of OVS.

	PSNR	SSIM	Crop	Distortion	Stability
Baseline	15.71	0.74	0.961	0.904	0.845
CoarseOnly	17.66	0.61	0.961	0.917	0.848
FineOnly	18.60	0.72	0.868	0.884	0.845
OVS	22.84	0.82	0.967	0.926	0.847

We select the SOTA image alignment method in [43] as a comparison to our OVS, which is a deep learning based method and specifically designed for adjacent frames alignment. In addition, we isolate the coarse alignment stage and the fine alignment stage from our OVS to compare their results with the complete version of OVS. Specifically, we choose DUT as the basic stabilizer and use the aligned frames from different models for warping. Their PSNR and SSIM metrics on the validation set are summarized in Table 2. It can be seen that 1) the baseline alignment method [43] can achieve good alignment results in terms of the SSIM scores and help DUT to improve the cropping ratio, demonstrating its effectiveness for image alignment. Our coarse alignment module can also help DUT improve the

Figure 6. Visual comparison of coarse alignment, fine alignment, and OVS. Red arrows indicate the misalignment.

cropping ratio and distortion metric as well as obtains a better PSNR score than [43]. However, its SSIM score is the worst among all the models, implying that there are some structural misalignment in the results. Our fine alignment module can obtain better out-of-boundary view synthesis results than the coarse alignment module in terms of both PSNR and SSIM metrics. However, it does not help DUT to achieve better stabilization results, implying that it can only serve as a refinement module. After combining both modules together, our OVS achieves the best view synthesis performance among all the models as well as significantly improve the performance of DUT for video stabilization.

Some visual results are shown in Figure 5 and Figure 6. The baseline method [43] produces significant discontinuities around the boundary of sharp edges and dynamic objects in its stabilized results, while our OVS has no such distortions. Similar discontinuities can be observed in the results of only using coarse alignment for warping, implying the necessity of fine alignment. However, only using fine alignment leads to less out-of-boundary view as shown by the large holes around the boundaries. These results confirm the complementarity between coarse alignment and fine alignment in our OVS, which help synthesize large out-of-boundary views with less distortions collaboratively.

4.4.2 Analysis of the Influence of Iterations

Table 3. The influence of different iterations.

Iteration	Crop	Distortion	Stability
0	0.867	0.895	0.845
5	0.938	0.930	0.847
10	0.967	0.926	0.847
15	0.970	0.925	0.847

We investigate the influence of iterations when employing OVS in warping-based stabilizers, *e.g.*, taking DUT as an exemplar stabilizer. The stability, distortion, and cropping ratio at different settings are shown in Table 3. It can be seen that as the number of iterations increases, the cropping ratio first increases and then almost reaches saturation. The distortion metric increases significantly at the beginning and then slightly decreases while the stability metric keep almost the same. As the number of iterations increases, more distant frames are aligned to the current reference frame for out-of-boundary view synthesis, leading to a larger area. Besides, there may be subtle misalignment accumulated during the iterations, especially misalignment from distant frames. Nevertheless, distant frames indeed contribute less to synthesis since they have less shared areas with the reference frame. Therefore, the distortion metric is slightly decreased as the number of iterations increases.

5. Limitation and discussion

Due to variations in adjacent frames, such as illumination, dynamic objects, and noise, some visible seams may exist in the stabilized frames, degrading the visual experience. Such discontinuities can be refined using an encoder-decoder network like [14]. It is noteworthy that we use the widely used PWCNet [30] for optical flow estimation following [5, 34] and an simple encoder-decoder network for fine alignment and affinity propagation, which can not be claimed as a major contribution. Actually, the major contrition of this work is that we provide a fresh perspective for warping-based methods towards full-frame video stabilization, *i.e.*, explicitly extrapolating the requisite out-of-boundary view during warping. In addition, we exploit the spatial coherence in the video to achieve this goal via a simple coarse-to-fine scheme. In the future, we plan to devise a more effective end-to-end model for better out-of-boundary view synthesis and video stabilization.

6. Conclusion

This paper presents a new Out-of-boundary View Synthesis (OVS) method that can help warping-based stabilizers to achieve near full frame stabilization with less distortions and better stability. OVS exploits the spatial coherence in the video to effectively align adjacent frames to reference frames and synthesize the out-of-boundary view, therefore benefiting the warping process in warping-based stabilizers directly by providing requisite candidate pixels. It can serve as a plug-and-play module and significantly improve both pixel- and grid-based warping stabilizers. We hope this study can provide valuable insights to the community and inspire follow-up research from a different but promising direction for video stabilization.

Acknowledgement Mr. Yufei Xu and Dr. Jing Zhang are supported by the ARC project FL-170100117.

References

[1] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3703–3712, 2019. 4

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision*, pages 404–417, 2006. 3

[3] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016. 1

[4] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2361–2379, 2020. 1, 5

[5] Jinsoo Choi and In So Kweon. Deep iterative frame interpolation for full-frame video stabilization. *ACM Transactions on Graphics*, 39(1), 2020. 1, 2, 4, 6, 7, 8

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016. 3

[7] Amit Goldstein and Raanan Fattal. Video stabilization using epipolar geometry. *ACM Transactions on Graphics*, 31(5):1–10, 2012. 1

[8] Matthias Grundmann, Vivek Kwatra, and Irfan Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 225–232, 2011. 1, 2

[9] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. 2000. 2, 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[11] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9000–9008, 2018. 4

[12] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep video inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5792–5801, 2019. 2

[13] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *ICLR 2015 : International Conference on Learning Representations 2015*, 2015. 6

[14] Victor Lempitsky, Andrea Vedaldi, and Dmitry Ulyanov. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018. 8

[15] Jingyuan Li, Fengxiang He, Lefei Zhang, Bo Du, and Dacheng Tao. Progressive reconstruction of visual structure for image inpainting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5962–5971, 2019. 2

[16] Sheng Li, Fengxiang He, Bo Du, Lefei Zhang, Yonghao Xu, and Dacheng Tao. Fast spatio-temporal residual network for video super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10522–10531, 2019. 2

[17] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 2

[18] Feng Liu, Michael Gleicher, Hailin Jin, and Aseem Agarwala. Content-preserving warps for 3d video stabilization. *ACM Transactions on Graphics*, 28(3):1–9, 2009. 1, 2

[19] Feng Liu, Michael Gleicher, Jue Wang, Hailin Jin, and Aseem Agarwala. Subspace video stabilization. *ACM Transactions on Graphics*, 30(1):1–10, 2011. 1, 2

[20] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. In *Advances in Neural Information Processing Systems*, volume 30, pages 1520–1530, 2017. 5

[21] Shuaicheng Liu, Ping Tan, Lu Yuan, Jian Sun, and Bing Zeng. Meshflow: Minimum latency online video stabilization. In *Proceedings of the European Conference on Computer Vision*, pages 800–815, 2016. 1, 2, 6, 7

[22] Shuaicheng Liu, Yinting Wang, Lu Yuan, Jiajun Bu, Ping Tan, and Jian Sun. Video stabilization with a depth camera. In *Proceedings of the European Conference on Computer Vision*, pages 89–95, 2012. 2

[23] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM Transactions on Graphics*, 32(4):1–10, 2013. 2, 6, 7

[24] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Steadyflow: Spatially smooth optical flow for video stabilization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4209–4216, 2014. 1, 2

[25] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 3

[26] Ty Nguyen, Steven W. Chen, Shreyas S. Shivakumar, Camillo Jose Taylor, and Vijay Kumar. Unsupervised deep homography: A fast and robust homography estimation model. In *IEEE Robotics and Automation Letters*, volume 3, pages 2346–2353, 2018. 3

[27] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9226–9235, 2019. 1

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. 4

[29] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2564–2571, 2011. 3

[30] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 4, 8

[31] Miao Wang, Guo-Ye Yang, Jin-Kun Lin, Song-Hai Zhang, Ariel Shamir, Shao-Ping Lu, and Shi-Min Hu. Deep online video stabilization with multi-grid warping transformation learning. *IEEE Transactions on Image Processing*, 28(5):2283–2292, 2018. 1, 2, 6, 7

[32] Xingkui Wei, Yinda Zhang, Zhuwen Li, Yanwei Fu, and Xiangyang Xue. Deepsfm: Structure from motion via deep bundle adjustment. In *European conference on computer vision*, pages 230–247. Springer, 2020. 2

[33] Sen-Zhe Xu, Jun Hu, Miao Wang, Tai-Jiang Mu, and Shi-Min Hu. Deep video stabilization using adversarial networks. In *Computer Graphics Forum*, volume 37, pages 267–276. Wiley Online Library, 2018. 1, 2

[34] Yufei Xu, Jing Zhang, Stephen J. Maybank, and Dacheng Tao. Dut: Learning video stabilization by simply watching unstable videos. *arXiv preprint arXiv:2011.14574*, 2020. 3, 4, 5, 6, 7, 8

[35] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *Proceedings of the European Conference on Computer Vision*, pages 467–483, 2016. 3

[36] Jiyang Yu and Ravi Ramamoorthi. Selfie video stabilization. In *Proceedings of the European Conference on Computer Vision*, pages 551–566, 2018. 1

[37] Jiyang Yu and Ravi Ramamoorthi. Robust video stabilization by optimization in cnn weight space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3800–3808, 2019. 1

[38] Jiyang Yu and Ravi Ramamoorthi. Learning video stabilization using optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8159–8167, 2020. 1, 6, 7

[39] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *European Conference on Computer Vision*, pages 528–543. Springer, 2020. 2

[40] Yibing Zhan, Jun Yu, Ting Yu, and Dacheng Tao. On exploring undetermined relationships for visual relationship detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5128–5137, 2019. 1

[41] Jing Zhang, Zhe Chen, and Dacheng Tao. Towards high performance human keypoint detection. *International Journal of Computer Vision*, pages 1–24, 2021. 1

[42] Jing Zhang and Dacheng Tao. Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal*, 8(10):7789–7817, 2020. 1

[43] Jirong Zhang, Chuan Wang, Shuaicheng Liu, Lanpeng Jia, Nianjin Ye, Jue Wang, Ji Zhou, and Jian Sun. Content-aware unsupervised deep homography estimation. In *European Conference on Computer Vision*, pages 653–669, 2020. 3, 7, 8

[44] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018. 2

[45] Minda Zhao and Qiang Ling. Pwstablenet: Learning pixel-wise warping maps for video stabilization. *IEEE Transactions on Image Processing*, 29:3582–3595, 2020. 6