DOI: 10.1111/cgf.13566
Pacific Graphics 2018
H. Fu, A. Ghosh, and J. Kopf
(Guest Editors)

Volume 37 (2018), Number 7

Deep Video Stabilization Using Adversarial Networks

Sen-Zhe Xu, Jun Hu, Miao Wang, Tai-Jiang Mu and Shi-Min Hu †

Department of Computer Science and Technology, Tsinghua University

Abstract

Video stabilization is necessary for many hand-held shot videos. In the past decades, although various video stabilization methods were proposed based on the smoothing of 2D, 2.5D or 3D camera paths, hardly have there been any deep learning methods to solve this problem. Instead of explicitly estimating and smoothing the camera path, we present a novel online deep learning framework to learn the stabilization transformation for each unsteady frame, given historical steady frames. Our network is composed of a generative network with spatial transformer networks embedded in different layers, and generates a stable frame for the incoming unstable frame by computing an appropriate affine transformation. We also introduce an adversarial network to determine the stability of a piece of video. The network is trained directly using the pair of steady and unsteady videos. Experiments show that our method can produce similar results as traditional methods, moreover, it is capable of handling challenging unsteady video of low quality, where traditional methods fail, such as video with heavy noise or multiple exposures. Our method runs in real time, which is much faster than traditional methods.

CCS Concepts

•***Computing methodologies*** → *Computer Graphics;*

1. Introduction

Video stabilization [MOG*06, CHA06, GKE11, GF12, LYTS13] is an important and widely studied problem in the community of computer vision. The goal of video stabilization is to generate a stable, visually-comfortable video from input video with jitters. In the past decades, masses of methods are proposed to solve this problem. The majority of the proposed methods tackle this problem via an off-line optimization, aiming at a smoothed camera path, to obtain a global view of the whole input video [LGJA09, LGW*11, GKE11, GF12, GKCE12, BAAR14, LYTS13]. Such methods are usually time-consuming. Meanwhile, only a few methods achieved online stabilization by estimating homography [YSCM06, BHL14, JWWY14] or transformation [LTY*16] between consecutive frames to smooth the camera motion. Although these methods can produce satisfying steady results, they would crash when the feature extraction is destroyed for video of low quality, such as heavy noise and multiple exposures. On the other hand, different transformations and explicit models designed to smooth the camera path always inherently define different undesired camera motions, which is hard to cover all the cases. In contrast with most of the methods aforementioned, we avoid defining jitter artificially, instead, come up with a deep framework to learn the unstable patterns in videos and remove them in an online and end-to-end fashion.

In recent years, deep convolutional neural networks have been widely used in fields of computer vision and graphics, which are proved to be efficient in most cases [KSH12, ZSQ*17, HGDG17, HZMH14, GEB16, KL17]. However, to our knowledge, hardly have there been deep learning methods for video stabilization. Video jitter is actually a disharmonious feeling perceived by human. Just like other defects of visual media such as blurry and compositing disharmony, which can be well removed by neural networks, it is reasonable that video jitter is also possible to be repaired by deep networks. The lack of deep video stabilization methods is mainly caused by two reasons, the shortage of supervision training data and the difficulty of problem definition specifically for convolutional neural networks.

To address this problem, we propose a novel deep framework for video stabilization. As to the training data problem, we choose to use the novel dataset provided by Wang *et al.* [WYL*18] recently. The dataset is collected through a well-designed hardware consists of two cameras, a standard hand-held camera and a camera with a pan-tilt stabilizer. The device can simultaneously shoot stable and unstable video pairs from real scenes. The stable and unstable frame pair is corresponding to each other only with a negligible parallax, and the transformation between them can be learned in a supervised way.

In order to solve the problem of video stabilization in an online manner, we proposed a generator-discriminator architecture to learn the video stabilization problem. We designed an encoder-decoder generator with spatial transformer networks

† S.-M. Hu is the corresponding author.

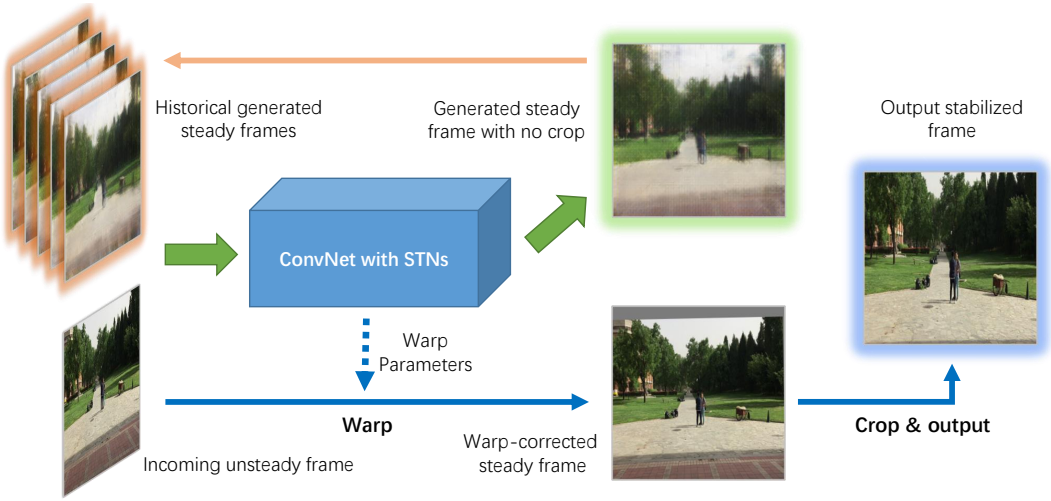

Figure 1: *The overview of our framework. Our network takes historical stabilized frames and the incoming unsteady frame as input. The output is a generated steady frame as historical condition and the corresponding transform parameters which are used to warp the unsteady input frame. The generated steady frame, containing sufficient features, is in turn appended at the historical stabilized frames. The final stabilization results are obtained by cropping the warped frame.*

(STNs) [HKW11, JSZK15] embedded in the different layers to predict the stable frames for unstable frames. Since full convolutional networks have weak ability to learn spatial transformations, the spatial transformation is entirely learned by the STNs. The encoder-decoder architecture can help those embedded STNs to be trained without any manually designed loss functions other than the similarity to the ground truth stable frame. Meanwhile, we defined a discriminator network to determine whether the generated frames are steady or not. The discriminator network somehow learns the human-like ability to distinguish stable or unstable videos, and helps the generator network to achieve a better ability of stabilization.

We test our method on various public videos and casually shot videos. Experiments show that our method can produce competitive results as the traditional ones, and runs in real-time at 30 fps which is much faster than off-line methods. Moreover, our method can run effectively on many types of low quality video cases, such as videos with heavy noise, multiple exposure videos or videos with periodic watermarks, where the traditional methods may fail.

2. Related Work

Our work aims to generate a visually stable, temporally consistent video from a jitter video in an adversarial way. This is closely related to the literature on existing video stabilization methods and deep image/video processing, including generative adversarial networks (GANs).

2.1. Video stabilization

Hand-held videos normally needs post-processing video stabilization techniques to remove large jitters. There is a rich history in digital video stabilization [MOG*06, CHA06, GKE11, GF12, LYTS13]. Most of the digital stabilization techniques estimate the camera trajectory from video content and then smooth it by removing the high-frequency component.

2D video stabilization methods estimate (bundled) homography or affine transformations between consecutive frames and smooth these transformations temporally. Pioneer works [MOG*06, CHA06] performs the low-pass filter on individual parameters to stabilize video content. Later, an $L1$-norm optimization based method [MOG*06] was proposed to synthesize camera path using simple partial camera paths. Bundled camera model [LYTS13] was introduced to optimize multiple local camera paths jointly. Recently, Zhang *et al.* [ZCKH17] proposed a method which optimizes geodesics on the Lie group embedded in transformation space.

3D-based stabilization methods perform 3D scene reconstruction [SSS06] to estimate camera trajectory. The first 3D stabilization method [LGJA09] was proposed by using content-preserve warping. Liu *et al.* [LGW*11] presented subspace video stabilization which smooths long tracked features under subspace constraints. Goldstein and Fattal [GF12] enhanced the length of feature trajectories with epipolar transfer. Bai *et al.* [BAAR14] proposed a semi-automatic stabilization algorithm which allows users to select proper feature trajectories. [GKCE12] addressed the rolling shutter issue in high-speed video.

In addition to the above methods, recently a 2D-3D mixed stabilization approach was proposed to stabilize 360 video [Kop16]. Generally, 2D stabilization methods work in a wider scope and efficiently, while 3D-based methods are able to produce better visual content.

Although previous global optimization methods have achieved state-of-the-art stabilization for videos, the computing process is usually off-line, which is not suitable for the popular live stream

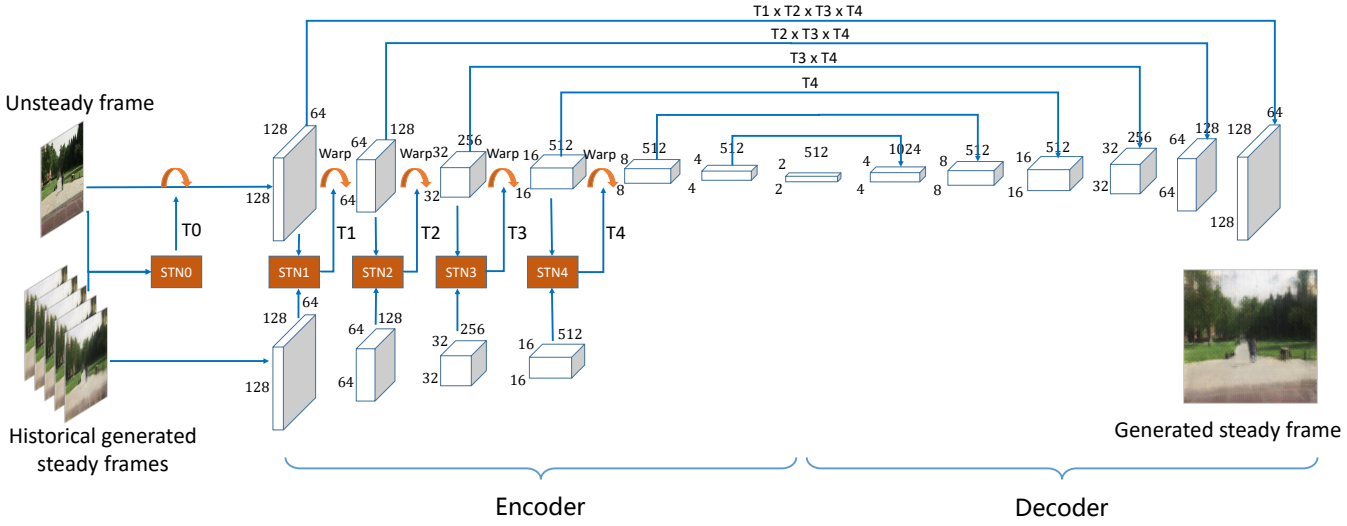

Figure 2: *The detailed learning network architecture of our proposed method.*

scenarios. Liu *et al.* [LTY*16] recently proposed an online video stabilization approach to compute warp functions for meshes of each incoming frame using historical camera path. Inspired by their work, we also present an online video stabilization method, using a generative adversarial network instead. Our approach just warps the unsteady video as closely to the steady one as possible without clearly computing a smooth camera path as what has been done in traditional feature based methods. This makes our method more robust to low quality videos, such as noise, blur, and multiple exposures, etc.

2.2. Deep video processing

In recent years, deep neural networks have been successfully applied to various computer vision tasks including recognition [KSH12, SZ14, HZRS16], segmentation [SLD17, ZSQ*17, HGDG17], recoloring [HZMH14], content generation [GEB16, IZZE17, ZPIE17] and image caption [VTBE15, KL17] etc, achieving comparative or even superior performance compared to traditional artificial algorithms. Considering the space and temporal consistency of videos, similar to some traditional 2D video applications, deep learning methods can also be exploited to camera pose estimation [NS17], action recognition [FPZ16, LWH*17, LSX*17, DSG17], deblurring [SDW*17, KLSH17], predicting optical flow [DFI*15, IMS*17], dynamic generation [VPT16, XWBF16] and frame synthesis [NML17, LYT*17] etc. Recently Wang *et. al.* [WYL*18] aimed to exploit a deep convolutional neural network for video stabilization.

To learn the temporal coherence among video frames, two or more consecutive videos frames are usually fed to convolutional neural networks, or frames could be fed to a recurrent neural network (RNN) [LWH*17, LSX*17] to learn the long-term dependencies. Our stabilization network also uses a recurrent structure to smooth the affine transformation in case of large jitters.

Generative adversarial network (GAN), which is composed of a generative network, called generator, and a discriminative one, called discriminator, was first proposed by Goodfellow *et al.* [GPAM*14] to generate a realistic version for an input noise image. The network is trained in an adversarial fashion by discriminating the faked version generated by the generator from the input ground truth till the discriminator can not tell the differences. Recently, GANs have been mainly used in various image content generation tasks [PKD*16, MML16, LTH*17, LLDX17, IZZE17, ZPIE17].

Our GAN for video stabilization does not directly generate a final steady pixel-wise image for each input unsteady frame; instead, the generated pixel-wise image is serving as the cropping free conditional input and the transformation parameters are gained to compute an online affine transformation for each input unsteady frame. The final steady video will be obtained by applying the online warp on each unsteady video frame.

3. Video stabilization network

In this section we describe the details of our proposed video stabilization network. Figure 1 shows the overview of our network. As our stabilization network works online, it only takes the historical stabilized frames and the incoming unsteady frame as input. The output is divided into two parts, a network-generated steady frame and the corresponding parameters of transformation, which is used to warp the input unsteady frame. The generated version of the steady frame is auto-completed by the network in the cropped area which is produced due to the stabilization process. The final stabilization results is obtained by cropping the warped frame. Particularly, we use the generated non-cropping frames as the subsequent steady inputs since they have the same size as the input frame and contain sufficient features. Before we go deeper into the network, we first introduce the training data.

3.1. Training data

Training data plays a key role on deep learning methods. In the problem of video stabilization, frame-by-frame correspondence unstable/stable video pairs are often rare to obtain. We use the training dataset proposed by Wang *et al.* [WYL*18] recently to train our network. This dataset contains 44 stable and unstable video pairs captured in the extensive outdoor scenes including road, buildings and vegetation. The correspondence of the video pairs is guaranteed by a well-designed hardware consisting of a normal hand-held camera and a camera with a pan-tilt stabilizer. Each video clip lasts 20-30 seconds or longer. When split into frames, the dataset gives more than 20,000 training samples.

3.2. Transform-aware encoder-decoder

Instead of directly learning the spatial transformation parameters at the end of a network, we predict the stabilized frame using an encoder-decoder framework in a generative manner. Different from most of encoder-decoder frameworks which can only handle tasks like pixel translation [IZZE17, ZPIE17], our framework need to be transform-aware. Figure 2 illustrates the architecture of our network. The encoder part of the network is basically composed of *conv* layers, of which different spatial transformation networks are placed in front or in middle. The decoder part is composed of *deconv* layers with skip connections to the corresponding *conv* layers.

Since a single unsteady frame is insufficient for the network to infer the stabilizing transformation, the inputs of our network include both the incoming unsteady frame I^t at time t and 5 stabilized sample frames evenly spaced during the last one second. Consider that our experimental video plays at 30 fps, we use $S^t = \left\{I_s^{t-7}, I_s^{t-13}, I_s^{t-19}, I_s^{t-25}, I_s^{t-31}\right\}$ as the conditional input frames at time t. S^t is converted to gray-scale before fed to the network and I^t retains RGB mode. So the total number of input channels is 8. As fully connected layers are contained in our work, all the inputs are resized to the size of 256×256 before fed into the network.

During the training phase, the conditional input S^t is replaced by the ground truth video frames $G^t = \left\{I_{gt}^{t-7}, I_{gt}^{t-13}, I_{gt}^{t-19}, I_{gt}^{t-25}, I_{gt}^{t-31}\right\}$, and the network's output is supervised by the ground truth steady frame I_{gt}^t.

I^t and S^t are firstly fed into STN_0 to perform an initial warp T_0 on I^t. The purpose of this step is to utilize the gradient backward propagated by the encoder to efficiently estimate a pre-warp of I^t as properly as possible. Then I^t and S^t are respectively pushed into parameter-shared *conv* layers to calculate their feature maps. These feature maps will be concatenated together only when they reach an inner STN. The reason why S^t and I^t's feature maps are not concatenated together is to ensure the effectiveness of training. Frames in S^t are much more similar to the ground truth I_{gt}^t than I^t since they are both steady. So the network tends to plagiarize frames in G^t rather than learn to transform I^t in the process of training if I^t and S^t's feature maps are not separated.

Each spatial transformation network STN_i consists of a light convolutional localization network to summarize the current feature map to the size of $4 \times 4 \times 16$, followed by a fully connected layer to regress the feature to a 2×3 affine transformation matrix T_i^t. Then warp is performed by the grid generator and sampler next using T_i^t. Note that the warp operation occurring in the inner block is also needed to be applied in the skip connections, since the feature map should be aligned. The cross multiplication from T_0^t to T_4^t are computed as the final transformation. We found that affine transformation is a proper choice to stabilize videos according to our experiments, and based on the traditional stabilization methods. In our case, affine transformation is more conducive to the convergence of network training. We also tried to use a homography transformation instead, however, there was no promotion of performance found.

The advantage of our encoder-decoder architecture over those learning the transformations directly at the end of the network is that our framework can make use of the information of each layer directly, so both the low-level and high-level feature correlations are considered to produce the final transformations. Our experiment also shows that a single ConvNet with the transformation only regressed at the end is hard to train for the spatial alignment task by optimizing the similarity loss like $L1$ or $L2$ distance. Extra conditional manual features, e.g., matched feature points distance, are required to guide the training. An insight of this phenomenon is that some low-level features are submerged in the deep layers and the correct transformation cannot be found just using the high-level features. Our network, on the contrary, can integrate multi-level cues while encoding the features and can be well trained directly just using the video pairs.

The output of the encoder-decoder network includes two parts, i.e, the predicted steady frame I_s^t generated by the decoder and the affine transformation T^t computed as the cross production from T_0^t to T_4^t orderly. We can warp the input I^t by T^t to get the warped version of the stabilized frame I_{warp}^t. I_s^t and I_{warp}^t have consistent content since the other the *conv* and *deconv* layers have little ability to learn spatial transfer, while I_{warp}^t is much more clear. However, I_s^t is useful as we choose it as the subsequent conditional input. I_s^t has size consistency to the former frames while still has strong features to analysis stabilization. The generated frames and corresponding warped frames are shown in Figure 3.

Our method improves [WYL*18]in the following aspects. First, the architecture of [WYL*18] is a single STN but with the localization network replaced by a ResNet50 model. It regresses warp parameters only at the end of the network. Our method utilizes a transform-aware encoder-decoder with multiple STNs to further support deeper feature map transformation. Second, [WYL*18] needs extra pre-computed matched feature points for training, which can have alignment errors due to the parallax. Their model is hard to converge without such kind of pre-computed feature matching. Our network does not require any pre-processing of hand-craft feature matching, since it directly learns how the generated frame is approaching the steady ground truth frame. Third, both methods need historical steady frames as conditional inputs, however the steady frames of [WYL*18] are warped frames with black borders. These black borders will disturb the network since they are changing. Our method takes the generated frames as steady frames where the black borders no longer exist.

Figure 3: *Illustration of generated frames and warped output frames for the input frames selected from three videos.*

3.3. Adversarial training

In this part we describe the training process of our network. Like lots of inharmonious factors, the video jitter is easily perceived by humans but difficult to be defined by the computer. Human can easily perceive the jitter of the video content even in poor picture quality such as heavy noise or blur, however many traditional stabilization methods will fail in these cases due to the loss of feature points. This observation inspires us to introduce a discriminator network to learn the human-like ability to perceive stable and unstable frames, and be adversarial with our encoder-decoder generator to help it to achieve a better ability of stabilization. Before we introduce the discriminator, we firstly talk about the training of the generator.

Thanks to the proposed encoder-decoder architecture, our generator network only needs the supervision of ground truth frame I_{gt}^t to learn the stabilization transformation. We make our generator's output to approximate the ground truth by $L1$ loss and the vgg19-net feature similarity. Although $L1$ loss is efficient, it can not capture the high-frequency part well. So we use the feature similarity output from a pre-trained vgg19-net as a reinforcement to the $L1$ loss. Finally, the stabilization loss is computed as:

$$L_{stab}(I^t, I_s^t) = \lambda_1 ||Vgg^{19}(I^t) - Vgg^{19}(I_s^t)|| + \lambda_2 ||I^t - I_s^t||, \quad (1)$$

where $\lambda_1 = 100$ and $\lambda_2 = 100$ are weighting parameters.

Fully convolutional networks(FCNs) has strong capacity to summarize patterns in the local area. Meanwhile, stability happen to be strongly related to the local change of an image. So we adopt an 8 layers fully convolutional network D_1 to discriminate the stability of a piece of video in training. D_1 has the same conditional input G^t as the generator. The loss function to train D_1 is designed as LSGAN [MLX*17], since the L2-form loss is proved to be more stable during training and generates higher quality results, as previous works demonstrated. The loss function computed as:

$$L_{D_1} = ||D_1(G^t, I_{gt}^t)||_2^2 + ||1 - D_1(G^t, I_s^t)||_2^2 \quad (2)$$

Temporal consistency is also guaranteed in the manner of adversarial training. Since our task is stabilization, the temporal consistency could be regard as the same matter. We adopt a same network D_2 as D_1 but have a conditional input of $A^t = \left\{I_s^{t-1}, I_s^{t-2}, I_s^{t-3}, I_s^{t-4}, I_s^{t-5}\right\}$ and to judge the stability of the adjacent stabilized frames. And make the generator to be adversarial with it. We also tried the Siamese framework to explicitly optimize the inter-frame difference, but got similar effect. The loss function to train D_2 is similar to D_1:

$$L_{D_2} = ||D_2(A^t, I_{gt}^t)||_2^2 + ||1 - D_2(A^t, I_s^t)||_2^2. \quad (3)$$

Be adversarial with D_1 and D_2, finally the generator's loss is:

$$L_G = L_{stab} + D_1(G^t, I_s^t) + D_2(A^t, I_s^t). \quad (4)$$

3.4. Implementation details

The activation functions used in our network are LeakyReLU with negative-slope set to 0.2 in the encoder and discriminator, and ReLU in the decoder except the last *deconv* layer. Weights are initialized according to a normal distribution (μ is 0 and σ is 0.02), while the bias of the STNs are set to identical transformations. Adam optimizer is used with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We trained the network for 40 epochs; each epoch has 30000 iterations. Batch-size is set to 1. The learning rate is set to $2e^{-4}$ initially, and linearly reduced to 0 in the last 20 epochs. In the test phase, we repeated the first frame with 30 times, and added these frames to the head of the video. These repeated frames serve as the historical steady frames.

4. Results and discussions

In this section, we first introduce the criterion used to evaluate the results of video stabilization. Then we perform ablation studies to validate our stabilization framework. After that we quantitatively compare our method against previous methods on a public videos set from [LYTS13] and conduct a user study to validate our approach with different effects imposed on videos captured by our own hand-held devices.

To make a quantitative evaluation, we follow the standards introduced in [LYTS13], namely, *cropping ratio, distortion and stability*. The stabilization results are considered to be good when the value of these metrics approaches 1. For clarity, we briefly explain these three quantitative metrics.

Cropping ratio measures the ratio of the area remained in the stabilization results after the black boundaries are cropped. A larger ratio means less original content cropping and hence better quality. The per-frame cropping ratio is the scale factor of homography between input and output frames during the stabilization. Cropping ratio of the whole video is averaged among all the frames of the video.

Distortion describes the degree of distortion of stabilization results compared to original ones. Distortion value for each frame is computed as the ratio of the two largest eigenvalues of the affine

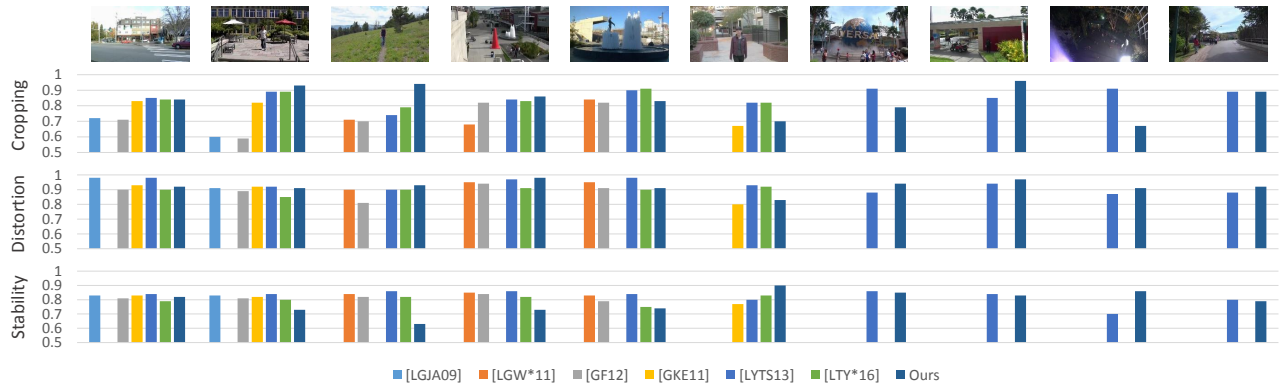

Figure 4: *Comparison with 10 publicly available videos in terms of three metrics: cropping ratio, distortion and stability.*

part of the homography. The smallest distortion value among all frames is defined as the distortion score of the whole video.

Stability evaluates how smooth a video is. Again, following [LYTS13], frequency-domain analysis of the camera path is used to compute the value of stability. Specifically, the rotation and translation sequences from all the homography transform between consecutive frames of the resulting video are regarded as two temporal sequences and the ratios of the lowest frequencies components(2nd to 6th) over the full frequencies (the DC component is excluded) are computed for the two sequences. The smaller ratio is regarded as the stability score of the stabilization results.

We select ten public videos from [LYTS13] as the test dataset for all the evaluations afterwards since these videos were commonly tested among previous methods [LGJA09,LGW*11,GF12,GKE11, LYTS13,LTY*16].

4.1. Ablation studies

Currently our network is trained with a hybrid of $L1$ loss, VGG loss, and spatial-temporal GAN loss. $L1$ loss is to make the generated image close to the ground truth, and the VGG loss is to make the generated image have a similar deep feature as the ground truth beyond appearance similarity (to better serve as historical steady input). The two GAN losses is to respectively ensure the generated frames to be equally distributed with the stable video frames in the long-time-range and the adjacent frame range. We study the effects of these losses by removing them severally. We have tried four configurations: 1) without L_{D_1}, 2) without L_{D_2}, 3) without $L_{D_1}+L_{D_2}$, 4) without VGG loss.

Table 1 shows the ablation studies for training losses. When discarding L_{D_1}, the results descends due to the lack of long-time-range temporal supervision. Things also happen when L_{D_2} is removed. We can also find that L_{D_1} affects distoration more while L_{D_2} affects cropping ratio more. When both L_{D_1} and L_{D_2} are removed, the situation is aggravated. We found that VGG loss also has an effect on the results. This is mainly because the VGG loss forces the generated image to be similar to the real steady frame, which makes it more suitable to be as a historical steady frame.

Table 1: *Ablation studies for training losses. Averaged cropping ratio, distortion, stability of w/o L_{D_1}, w/o L_{D_2}, w/o $L_{D_1}+L_{D_2}$, w/o Vgg and Ours are listed.*

Method	cropping ratio	distortion	stability
w/o L_{D_1}	0.7870	0.8022	0.8520
w/o L_{D_2}	0.6936	0.8485	0.8686
w/o $L_{D_1}+L_{D_2}$	0.7339	0.8303	0.8350
w/o Vgg	0.7598	0.8365	0.8497
Ours	0.8221	0.9022	0.8488

In order to make the features free to transform in arbitrary encoding layers of the network, we add STNs in deeper layers of the network. Since the spatial dimensions of feature maps from the last 3 *conv-deconv* blocks are too small, we did not use STNs in the inner-most three *conv-deconv* blocks. To explore how each STN impacts the network output, we drop each of the STNs respectively.

Table 2: *Ablation studies for STN layers. Averaged cropping ratio, distortion, stability of w/o STN_0, w/o STN_1, w/o STN_2, w/o STN_3, w/o STN_4 and Ours are listed.*

Method	cropping ratio	distortion	stability
w/o STN_0	0.8044	0.8880	0.8581
w/o STN_1	0.8082	0.8991	0.8542
w/o STN_2	0.8174	0.9172	0.8559
w/o STN_3	0.8180	0.9193	0.8485
w/o STN_4	0.8189	0.9205	0.8435
Ours	0.8221	0.9022	0.8488

Table 2 shows the ablation studies for STN layers. Basically we can find that only removing one STN has similar effect on the results. That is because while removing one STN, the role of the STN will be replaced by the STNs of other layers to a certain degree in the training phase.

We also studied the effect of the number of conditional input frames on the results. Currently we select 5 historical stabilized frames equally spaced during the last one second serving as our

conditional inputs. The choice of the number of frames is empirical, as we believe the last one second is a proper time span to infer the stabilizing transformation, and the input feature thickness is appropriate for training. In order to study the influence of the conditional inputs, we fed the network with less or more previous frames with the same interval as our conditional inputs.

Table 3 shows the ablation studies for the conditional input. Basically, we can find that the more the conditional frames input, the better the result is. This is not surprising since more frame means stronger temporal supervision and more information. However, more frames also make the feature map bloated and decrease the convergence speed of the model training. In our experiment, when the number of conditional input frames exceeds 5, the result promotion becomes small.

Table 3: *Ablation studies for the conditional inputs. Averaged cropping ratio, distortion, stability of with less frames, with more frames and Ours are listed.*

Method	cropping ratio	distortion	stability
with less frames	0.6435	0.9143	0.8382
with more frames	0.8120	0.9390	0.8558
Ours	0.8221	0.9022	0.8488

4.2. Quantitative evaluation

We compare our online learning method with both traditional offline methods [LGJA09,LGW*11,GF12,GKE11,LYTS13] and online method [LTY*16].

The detailed data are shown in figure 4, based on the results provided by the corresponding authors or found on their project pages (missing results are left blank). When compared to the state-of-the-art online method [LTY*16], we can see from the first 6 videos, overall, our method performs better under the cropping ratio and distortion metrics. This is because of the Meshflow [LTY*16] method computed warp functions for meshes of the frame while our method predicts an affine transform for each frame, i.e. regarding the full resolution of the frame as a single mesh. So, our method would ignore some detailed local smoothness during stabilization, which in turn keeps a larger cropping ratio and less distortion. Comparing with offline optimization methods seems a little unfair for our method since the future frames are not available for stabilizing the current frame. As a result, the stability score of our method is lower than those methods. This can be further demonstrated on a category-wise comparison against state-of-the-art offline method [LYTS13] in figure 5, where we select 3 videos for each category (including *Regular, Quick Rotation, Quick Zooming, Parallax, Running and Crowd*), classified in terms of scene type and camera motion ,from the publicly available video set [LYTS13]. It can be drawn from this figure that our method achieve a slightly better results only among videos with quick rotations. This might merely be the reason that our learning network has seen such quick rotation videos during the training process before.

Overall, although our online stabilization learning framework obtains lower stability than offline methods or state-of-the-art online method inevitably, our method can run faster than all these methods, and the averaged running time is given in Table 4.

Table 4: *Running time performance. The FPS(frames per second) of typical offline and online methods are listed.*

Method	FPS
Bundle Camera [LYTS13]	3.5
MeshFlow [LTY*16]	22.0
Ours	30.1

4.3. User study

In order to validate the robustness of our method when the features of the frame content are difficult to be reliably tracked for some low-quality videos. Here, we introduce 4 common kinds of low quality videos: *Camera lens blur* is commonly noticed in pinhole cameras, where objects away from the focal plane will be blurred; *noise* videos are easy to be captured when the lighting condition becomes dim; *multiple exposures* would result in double vision or ghosting; *watermarks* are commonly used for Internet videos aiming for copyright protection. These effects would cause the feature tracking procedure to be interrupted frequently or even to fail. Figure 6 presents the 4 kinds of low quality video frame, and we use Gaussian noise to demonstrate the *noise* effect.

We captured 4 casual videos and each was applied with the aforementioned effects, resulting in 16 low quality videos.The three quantitative metrics mentioned above are estimated by homography matching, and when evaluating low quality videos, the feature matching fails. So we conducted a user study to evaluate video stability. We compare our method against the commercial offline stabilization software *Adobe Premier Pro CC 2018*. As far as we know, the method described in [LGW*11] was incorporated into the Adobe Premiere stabilizer. All the low quality videos were fed to both Adobe Premiere stabilizer and our method to generate the final results. However, Adobe Premiere stabilizer failed to generate results for 2 videos with heavy Gaussian noise, which were eliminated from the user study. 34 people recruited from the campus were asked to figure out which video seemed more stable for the randomly permuted 14 pairs of stabilization results, or indicate a 'indistinguishable', regardless of cropping ratio or sharpness.

The averaging percent of choices among all the participants for each kind of low quality effect was shown in figure 7. As can be seen from the figure, all the participants picked the results of our method as the more stable, except the camera lens blur effect, which basically agreed with our discussions. Our method does not explicitly extract feature points for path estimation and smoothing, so the failure of feature point extraction or matching in low-quality videos does not impact on our approach.

4.4. Limitations

Our current video stabilization learning network has its own limitations. First, the network only generated a global affine transformation for each video frame, which omitted the local transformations.

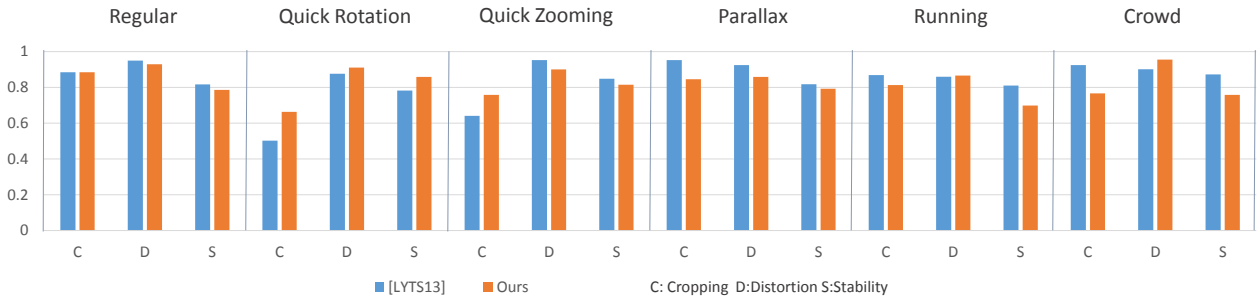

Figure 5: *Comparison with state-of-the-art offline method in different categories.*

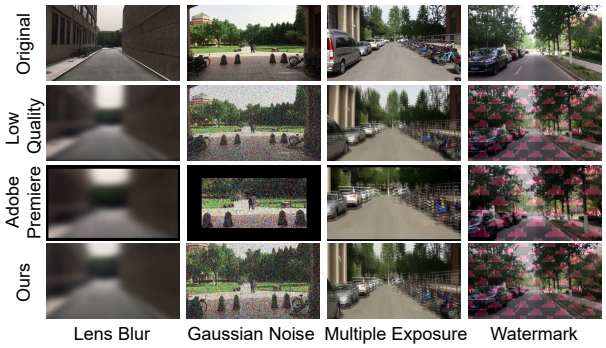

Figure 6: *Low quality effects. From top to bottom: original frame, frame with low quality effect, result frame of Adobe Premier stabilizer, result frame of our method.*

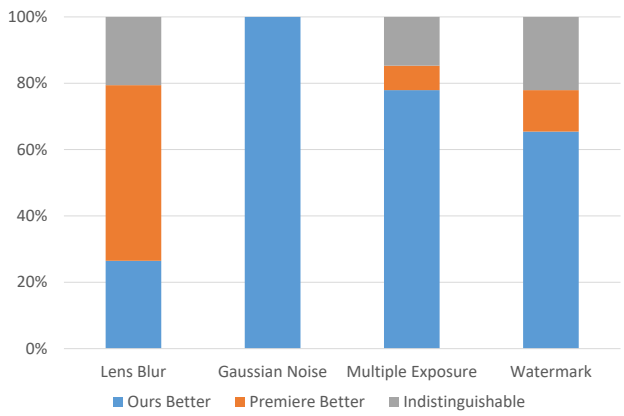

Figure 7: *User study results comparing Adobe Premier stabilizer with our methods under different low quality effects.*

Dividing the video frame into meshes as in [LTY*16] and learning transformations on these smaller meshes seems to be a promising approach. Second, the generated affine transformation only considers transformation from the previous frame, which results in a weak temporal coherence. A more complex RNN could be tried in the future to learn the long term dependencies in the temporal domain.

5. Conclusions

In this paper, we proposed to solve the traditional video stabilization problem using a novel online GANs. This learning network regarded the video stabilization as an affine transformation generation between consecutive video frames instead of smoothing a camera path as in traditional feature tracking based methods. The experiments demonstrated that our method was comparable to current state-of-the-art online methods on a public video set and more suitable for low quality videos, especially when the feature tracking is unreliable or impossible.

Acknowledgements

The authors would like to thank all the reviewers. This work was supported by the National Natural Science Foundation of China (Project Number 61561146393 and 61521002) and China Postdoctoral Science Foundation (Project Number 2016M601032).

References

[BAAR14] BAI J., AGARWALA A., AGRAWALA M., RAMAMOORTHI R.: User-assisted video stabilization. In *Proceedings of the 25th Eurographics Symposium on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2014), EGSR '14, Eurographics Association, pp. 61–70. 1, 2

[BHL14] BAE J., HWANG Y., LIM J.: Semi-online video stabilization using probabilistic keyframe update and inter-keyframe motion smoothing. In *2014 IEEE International Conference on Image Processing (ICIP)* (Oct 2014), pp. 5786–5790. 1

[CHA06] A robust real-time video stabilization algorithm. *Journal of Visual Communication and Image Representation 17*, 3 (2006), 659 – 673. 1, 2

[DFI*15] DOSOVITSKIY A., FISCHER P., ILG E., HÃĎUSSER P., HAZIRBAS C., GOLKOV V., V. D. SMAGT P., CREMERS D., BROX T.: Flownet: Learning optical flow with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)* (Dec 2015), pp. 2758–2766. 3

[DSG17] DIBA A., SHARMA V., GOOL L. V.: Deep temporal linear encoding networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), pp. 1541–1550. 3

[FPZ16] FEICHTENHOFER C., PINZ A., ZISSERMAN A.: Convolutional two-stream network fusion for video action recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016), pp. 1933–1941. 3

[GEB16] GATYS L. A., ECKER A. S., BETHGE M.: Image style transfer

using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016), pp. 2414–2423. 1, 3

[GF12] GOLDSTEIN A., FATTAL R.: Video stabilization using epipolar geometry. *ACM Trans. Graph. 31*, 5 (Sept. 2012), 126:1–126:10. 1, 2, 6, 7

[GKCE12] GRUNDMANN M., KWATRA V., CASTRO D., ESSA I.: Calibration-free rolling shutter removal. In *International Conference on Computational Photography [Best Paper]* (2012). 1, 2

[GKE11] GRUNDMANN M., KWATRA V., ESSA I.: Auto-directed video stabilization with robust l1 optimal camera paths. In *Proc. Int. Conf. CVPR* (2011), IEEE, pp. 225–232. 1, 2, 6, 7

[GPAM*14] GOODFELLOW I. J., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A., BENGIO Y.: Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (Cambridge, MA, USA, 2014), NIPS'14, MIT Press, pp. 2672–2680. 3

[HGDG17] HE K., GKIOXARI G., DOLLÃAR P., GIRSHICK R.: Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)* (Oct 2017), pp. 2980–2988. 1, 3

[HKW11] HINTON G. E., KRIZHEVSKY A., WANG S. D.: Transforming auto-encoders. In *Proceedings of the 21th International Conference on Artificial Neural Networks - Volume Part I* (Berlin, Heidelberg, 2011), ICANN'11, Springer-Verlag, pp. 44–51. 2

[HZMH14] HUANG H.-Z., ZHANG S.-H., MARTIN R. R., HU S.-M.: Learning natural colors for image recoloring. *Comput. Graph. Forum 33*, 7 (Oct. 2014), 299–308. 1, 3

[HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016), pp. 770–778. 3

[IMS*17] ILG E., MAYER N., SAIKIA T., KEUPER M., DOSOVITSKIY A., BROX T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), pp. 1647–1655. 3

[IZZE17] ISOLA P., ZHU J. Y., ZHOU T., EFROS A. A.: Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), pp. 5967–5976. 3, 4

[JSZK15] JADERBERG M., SIMONYAN K., ZISSERMAN A., KAVUKCUOGLU K.: Spatial transformer networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2* (Cambridge, MA, USA, 2015), NIPS'15, MIT Press, pp. 2017–2025. 2

[JWWY14] JIANG W., WU Z., WUS J., YU H.: One-pass video stabilization on mobile devices. In *Proceedings of the 22Nd ACM International Conference on Multimedia* (New York, NY, USA, 2014), MM '14, ACM, pp. 817–820. 1

[KL17] KARPATHY A., LI F.-F.: Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence 39*, 4 (April 2017), 664–676. 1, 3

[KLSH17] KIM T. H., LEE K. M., SCHÃŰLKOPF B., HIRSCH M.: Online video deblurring via dynamic temporal blending network. In *2017 IEEE International Conference on Computer Vision (ICCV)* (Oct 2017), pp. 4058–4067. 3

[Kop16] KOPF J.: 360 video stabilization. *ACM Trans. Graph. 35*, 6 (Nov. 2016), 195:1–195:9. 2

[KSH12] KRIZHEVSKY A., SUTSKEVER I., HINTON G. E.: Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (USA, 2012), NIPS'12, Curran Associates Inc., pp. 1097–1105. 1, 3

[LGJA09] LIU F., GLEICHER M., JIN H., AGARWALA A.: Content-preserving warps for 3d video stabilization. *ACM Trans. Graph. 28*, 3 (2009), 44:1–9. 1, 2, 6, 7

[LGW*11] LIU F., GLEICHER M., WANG J., JIN H., AGARWALA A.: Subspace video stabilization. *ACM Trans. Graph. 30*, 1 (2011), 4:1–10. 1, 2, 6, 7

[LLDX17] LIANG X., LEE L., DAI W., XING E. P.: Dual motion gan for future-flow embedded video prediction. In *2017 IEEE International Conference on Computer Vision (ICCV)* (Oct 2017), pp. 1762–1770. 3

[LSX*17] LIU J., SHAHROUDY A., XU D., CHICHUNG A. K., WANG G.: Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017), 1–1. 3

[LTH*17] LEDIG C., THEIS L., HUSZÃAR F., CABALLERO J., CUNNINGHAM A., ACOSTA A., AITKEN A., TEJANI A., TOTZ J., WANG Z., SHI W.: Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), pp. 105–114. 3

[LTY*16] LIU S., TAN P., YUAN L., SUN J., ZENG B.: Meshflow: Minimum latency online video stabilization. In *Computer Vision – ECCV 2016* (Cham, 2016), Leibe B., Matas J., Sebe N., Welling M., (Eds.), Springer International Publishing, pp. 800–815. 1, 3, 6, 7, 8

[LWH*17] LIU J., WANG G., HU P., DUAN L. Y., KOT A. C.: Global context-aware attention lstm networks for 3d action recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), pp. 3671–3680. 3

[LYT*17] LIU Z., YEH R. A., TANG X., LIU Y., AGARWALA A.: Video frame synthesis using deep voxel flow. In *2017 IEEE International Conference on Computer Vision (ICCV)* (Oct 2017), pp. 4473–4481. 3

[LYTS13] LIU S., YUAN L., TAN P., SUN J.: Bundled camera paths for video stabilization. *ACM Trans. Graph. 32*, 4 (July 2013), 78:1–78:10. 1, 2, 5, 6, 7

[MLX*17] MAO X., LI Q., XIE H., LAU R. Y., WANG Z., SMOLLEY S. P.: Least squares generative adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on* (2017), IEEE, pp. 2813–2821. 5

[MML16] MICHAEL MATHIEU C. C., LECUN Y.: Deep multi-scale video prediction beyond mean square error. In *International Conference on Learning Representations 2016(ICLR)* (2016). 3

[MOG*06] MATSUSHITA Y., OFEK E., GE W., TANG X., SHUM H.-Y.: Full-frame video stabilization with motion inpainting. *IEEE Trans. Pattern Anal. Machine Intell. 28*, 7 (2006), 1150–1163. 1, 2

[NML17] NIKLAUS S., MAI L., LIU F.: Video frame interpolation via adaptive separable convolution. In *2017 IEEE International Conference on Computer Vision (ICCV)* (Oct 2017), pp. 261–270. 3

[NS17] NAKAJIMA Y., SAITO H.: Robust camera pose estimation by viewpoint classification using deep learning. *Computational Visual Media 3*, 2 (Jun 2017), 189–198. 3

[PKD*16] PATHAK D., KRÃĎHENBÃIJHL P., DONAHUE J., DARRELL T., EFROS A. A.: Context encoders: Feature learning by inpainting. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016), pp. 2536–2544. 3

[SDW*17] SU S., DELBRACIO M., WANG J., SAPIRO G., HEIDRICH W., WANG O.: Deep video deblurring for hand-held cameras. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), pp. 237–246. 3

[SLD17] SHELHAMER E., LONG J., DARRELL T.: Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 39*, 4 (April 2017), 640–651. 3

[SSS06] SNAVELY N., SEITZ S. M., SZELISKI R.: Photo tourism: Exploring photo collections in 3d. *ACM Trans. Graph. 25*, 3 (2006), 835–846. 2

[SZ14] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. *CoRR abs/1409.1556* (2014). 3

[VPT16] VONDRICK C., PIRSIAVASH H., TORRALBA A.: Generating videos with scene dynamics. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (USA, 2016), NIPS'16, Curran Associates Inc., pp. 613–621. 3

[VTBE15] VINYALS O., TOSHEV A., BENGIO S., ERHAN D.: Show and tell: A neural image caption generator. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015), pp. 3156–3164. 3

[WYL*18] WANG M., YANG G., LIN J., SHAMIR A., ZHANG S., LU S., HU S.: Deep online video stabilization. *arXiv preprint arXiv:1802.08091* (2018). 1, 3, 4

[XWBF16] XUE T., WU J., BOUMAN K. L., FREEMAN W. T.: Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (USA, 2016), NIPS'16, Curran Associates Inc., pp. 91–99. 3

[YSCM06] YANG J., SCHONFELD D., CHEN C., MOHAMED M.: Online video stabilization based on particle filters. In *2006 International Conference on Image Processing* (Oct 2006), pp. 1545–1548. 1

[ZCKH17] ZHANG L., CHEN X.-Q., KONG X.-Y., HUANG H.: Geodesic video stabilization in transformation space. *Trans. Img. Proc. 26*, 5 (May 2017), 2219–2229. 2

[ZPIE17] ZHU J. Y., PARK T., ISOLA P., EFROS A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)* (Oct 2017), pp. 2242–2251. 3, 4

[ZSQ*17] ZHAO H., SHI J., QI X., WANG X., JIA J.: Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), pp. 6230–6239. 1, 3