GlobalFlowNet: Video Stabilization using Deep Distilled Global Motion Estimates

Jerin Geo James Devansh Jain Ajit Rajwade
Indian Institute of Technology Bombay
{jeringeo,devanshdvj,ajitvr}@cse.iitb.ac.in

Abstract

Videos shot by laymen using hand-held cameras contain undesirable shaky motion. Estimating the global motion between successive frames, in a manner not influenced by moving objects, is central to many video stabilization techniques, but poses significant challenges. A large body of work uses 2D affine transformations or homography for the global motion. However, in this work, we introduce a more general representation scheme, which adapts any existing optical flow network to ignore the moving objects and obtain a spatially smooth approximation of the global motion between video frames. We achieve this by a knowledge distillation approach, where we first introduce a low pass filter module into the optical flow network to constrain the predicted optical flow to be spatially smooth. This becomes our student network, named as GLOBALFLOWNET*. Then, using the original optical flow network as the teacher network, we train the student network using a robust loss function. Given a trained* GLOBALFLOWNET*, we stabilize videos using a two stage process. In the first stage, we correct the instability in affine parameters using a quadratic programming approach constrained by a user-specified cropping limit to control loss of field of view. In the second stage, we stabilize the video further by smoothing global motion parameters, expressed using a small number of discrete cosine transform coefficients. In extensive experiments on a variety of different videos, our technique outperforms state of the art techniques in terms of subjective quality and different quantitative measures of video stability. Additionally, we present a new measure for evaluation of video stabilization based on the flow generated by* GLOBALFLOWNET *and argue that it is based on a more general motion model in contrast to the affine motion model on which most existing measures are based. The source code is publicly available at https://github.com/GlobalFlowNet/GlobalFlowNet*

1. Introduction

Videos acquired by amateur photographers or lay users from hand-held cameras or mobile phones are subject to a large magnitude of undesirable and discontinuous motion. The process of eliminating or reducing this undesirable motion is called video stabilization. In some setups, the camera can be mounted on stable stands or dollies, but this is infeasible in many commonplace scenarios. Some cameras are equipped with hardware such as gyroscopes for stabilization, but the state of the art in video stabilization still adopts software-based approaches due to the gyroscope's cost, weight and error-pone motion estimation [20, 23]. Apart from casual hand-held photography, the need for video stabilization also arises in endoscopy [10], underwater imaging [21] and aerial photography from drones/helicopters [11]. Many video stabilization techniques consist of three broad steps: (1) estimation of the motion between consecutive or temporally neighboring frames assuming a suitable motion model, (2) temporal motion smoothing assuming an appropriate motion model for the underlying stable video, and (3) re-targeting or warping of the frames of the unstable video so as to generate a stabilized video. There exists a large body of literature on video stabilization, with differences in the way these three steps are executed. Several of these techniques are summarized below.

Related work (Classical Approaches): Many traditional techniques assume that the motion between consecutive frames can be approximated using 2D affine transformations or homographies [19, 6], and seek to smooth a sequence of such parameters to render a stabilized video. For computing the parameterized motion, many of these techniques make use of robust point tracking methods [19, 6, 15]. However, 2D motion models cannot accurately account for the motion between consecutive video frames for scenes with significant depth variation or significant camera motion. Some methods such as [15] approximate the motion between consecutive frames by means of patch-wise 2D models or homographies. The method in [17] performs three tasks in an iterated fashion: determining the smooth global background motion between consecutive frames by detecting moving objects, inpainting the flow in those regions, and smoothing per-pixel optical flow vectors across time. There also exist methods which make use of epipolar geometry [4] or various geometrically motivated subspace constraints [13]. The latter technique requires fairly long feature tracks which may not be available in many real-world videos. Finally, many techniques which use 3D information have also been proposed, for example methods that use structure from motion [12], a depth camera [26] or a light field camera [24].

Related work (Deep Learning Approaches): Deep neural network (DNN) based approaches for video stabilization have become very popular in recent years. The work in [35] represents the warp fields using the weights of an unsupervised DNN, which minimizes the sum of two terms: a regularizer that encourages the warp fields to be piecewise linear, and a fidelity term which minimizes the distance between corresponding pixels in consecutive frames in the stabilized video. This approach, though elegant, must evolve the network weights afresh for every video, and has very high computational cost. The work in [36] trains a DNN offline on a large video dataset with synthetic unstable motion. In an unsupervised fashion, the weights of the DNN are evolved so as to generate warp fields that (1) have dominant low-frequency content in the Fourier domain, and (2) yield minimal distance between corresponding pixels in consecutive frames of the stabilized video. The work uses frame-to-frame optical flow as initial input and requires a number of pre-processing steps to: (1) identify regions with moving objects from the optical flow fields using a variety of segmentation masks for typical foreground objects obtained from [39], (2) identify regions of inaccurate optical flow, and (3) inpaint all such regions using the PCA-based approach from [32]. The work in [2] trains two DNNs for performing video stabilization via frame interpolation to smooth the motion between consecutive frames. The $(i-1)^{\text{th}}$ and $(i+1)^{\text{th}}$ frames are linearly warped mid-way toward each other using the bidirectional optical flow between them. The resulting warped frames are passed through a U-Net [22] to generate the i^{th} intermediate 'stabilized' frame. This interpolation process is carried out iteratively which may accumulate blur. To prevent this, the intermediate stabilized frames are also passed through a Resnet [8]. The motion smoothing in this approach is always linear without any adaptation of the smoothing parameters to the motion at different time instants or at different depths. Similar in spirit to [2], work in [18, 34] perform full-frame video stabilization by bringing in border-based frame inpainting. However, the approach in [18] is computationally expensive. The approach in [30] trains a Siamese network to generate a warp grid for video stabilization using stable and unstable video pairs from the Deepstab dataset [33]. Their approach is based purely on color without using any motion parameters and does not perform very well. The approach in [33] uses spatial transformer networks along with adversarial networks for video stabilization, but suffers from problems due to inadequate training data. The work in [37], which is called PWStableNet, uses a supervised training approach based on a cascade of encoder-decoder units to optimize a combination of criteria such as fidelity w.r.t. the underlying stable video, and various motion and feature-based characteristics. This approach has limitations in terms of training data scarcity and generalizability.

Overview of Proposed Approach: A major contribution of our work is a novel method of estimating the global motion between frames of an unstable video, proposed in Sec. 2. Our method involves training a network GLOBALFLOWNET in a teacher-student fashion in such a way that it imposes a smooth and compact representation for the global motion and is designed to not be influenced by the motion in regions containing moving objects. Our method yields global motion representations that are more general than 2D affine or homography transformation and does not require any salient feature point tracking. Given a pre-trained GLOBALFLOWNET, we achieve video stabilization using a two-stage process comprising a novel global affine parameter smoothing step (Sec. 3.1) and a novel residual level smoothing step (Sec. 3.2) involving low frequency discrete cosine transform (DCT) coefficients of the residual flow. Both these steps smooth the parameters in the temporal direction. The first step acts as a very useful initial condition, whereas the second step is necessary to significantly improve stabilization performance. This is because it works with a global motion model that despite being very compact, is much more general than just 2D affine or homography. Our overall approach for video stabilization is simple, computationally efficient and interpretable. In extensive experiments (see Sec. 4), it outperforms state of the art techniques in terms of stability measures. We also propose a new video stabilization measure which uses the low frequency representation from Sec. 2.2 to quantify the temporal smoothness of the global motion between successive pairs of frames. Our measure uses a more general motion model than existing measures which largely use affine transformations.

2. Global Motion Estimation

A key step in video stabilization is the estimation of global motion between consecutive video frames (or temporally nearby video frames), followed by temporal smoothing of the motion parameters. The difference between the original global motion and the global motion in a stabilized video constitutes the *warp field*, which when applied to the unstable frames, stabilizes the video. An ideal global motion or warp field will contain motion discontinuities due to

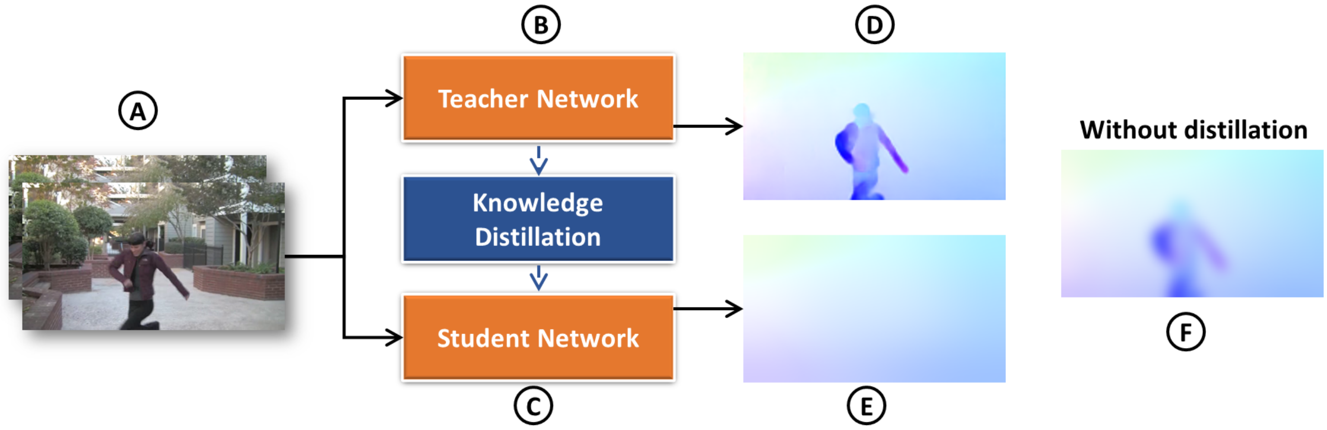

Figure 1: Network architecture for our knowledge distillation approach: a teacher component (B) based on PWC-Net [25] which produces inter-frame motion estimate (D), and a modified student architecture (C) that obtains smooth global inter-frame motion (E) after training with a robust loss). (F) represents a flow obtained by just low-pass filtering of the flow in (D) without knowledge distillation; it is not part of the network and is shown only for comparison.

scene depth variations or occlusions. If this warp field is applied directly, it would create holes in the resulting images, requiring non-trivial, potentially error-prone inpainting operations. Hence, many methods approximate the warp fields using continuous motion vector fields or parametrically via 2D affine transformations or homography [6, 19]. In this work, we aim to train a network to acquire an inherent ability to produce smooth global motion between consecutive frames in a manner not influenced by independently moving objects in the scene. We achieve this using a knowledge distillation mechanism sketched in Fig. 1.

2.1. Knowledge Distillation Approach

Given a standard optical flow estimation network $\mathcal{T}$ which acts as a teacher, we create a student network $\mathcal{S}$, initialized with the weights of $\mathcal{T}$, by introducing a *low-pass filter module* (LPM). The LPM is chosen in a manner such that it can represent the global motion between two frames with a high degree of accuracy, but not the motion in regions containing independently moving objects. This is because video stabilization is expected to smooth *only* global inter-frame motion and *not* cause any change to the motion of independently moving objects. Without any further training of $\mathcal{S}$, the optical flow $\boldsymbol{f_S}$ produced by $\mathcal{S}$, would be a blurred version of the optical flow $\boldsymbol{f_T}$ produced by $\mathcal{T}$, as shown in Fig. 1(F). Such a flow would necessarily contain components of the motion from the independently moving objects leaked into the neighbouring pixels, which would create motion artifacts if used for video stabilization. Instead, we would want $\mathcal{S}$ to mimic the optical flow produced by $\mathcal{T}$ in all regions except the moving objects. This can be achieved by training $\mathcal{S}$ using a robust loss function given as:

$$C_1 = \sum_{l=1}^{N_T}\sum_{i=1}^{N} \mathfrak{R}\left(\|\boldsymbol{f_T}(l,i) - \boldsymbol{f_S}(l,i)\|_2\right), \tag{1}$$

$$\mathfrak{R}(x;\alpha,c) = \frac{|\alpha-2|}{\alpha}\left(\left(\frac{(x/c)^2}{|\alpha-2|}+1\right)^{\alpha/2}-1\right), \tag{2}$$

where N is the number of pixels per image, N_T is the number of image-pairs on which the network is trained, and i, l are indices for pixels and image-pairs respectively. The robust loss $\mathfrak{R}(.)$ that we use here was introduced in [1], for which we chose shape parameter $\alpha \triangleq -0.1$ and scale parameter $c \triangleq 0.001$ in our work.

This approach of (1) constraining the dimensionality of $\boldsymbol{f_S}$, and (2) using a robust loss as opposed to a squared error, ensures that the training of $\mathcal{S}$ focuses on global motion and is not influenced by the flow on moving objects. For the student network $\mathcal{S}$, which we henceforth refer to as GLOBALFLOWNET, we use the well known PWC-Net architecture for optical flow [25]. For the low pass filter module LPM, we use low frequency (upto some cutoff frequency) DCT basis vectors.

2.2. Low Pass Filter Module

As mentioned before, LPM should be able to represent global motion with high accuracy and should exhibit poor accuracy in regions with moving objects. With this aim in mind, any low rank off-the-shelf frequency transforms could be used for LPM. In our method, we choose low frequency components of the Discrete Cosine Transform (DCT). This choice is motivated by our experiments performed on a large video dataset such as [16]. For our experiments, we selected pairs of consecutive frames that did not contain independently moving objects. For such frames, we observed that the optical flow, which is the same as the low-frequency global motion, is accurately expressed using *a very small number of DCT coefficients* with frequencies ranging from $(u, v) = (0, 0)$ to $(u, v) = (R, R)$ for a cutoff frequency of $R \leq 8$, as can be seen in Figure 1 of the suppl. material at [27].

2.3. GlobalFlowNet: Global Motion Estimation Network

The PWC-Net architecture [25], on which GLOBALFLOWNET is based, has three important modules, as illustrated in supplementary material [27]:

1. *Feature extractor*: This module converts the original image into a feature map in each level of refinement.

2. *Warping layer*: At each level, there is an estimate of the optical flow from previous level. The warping layer warps the features of the target image based on this flow. This layer helps in obtaining the optical flow for small and fast moving objects accurately.

3. *Cost volume and context network*: The warped target feature-map is correlated with the source feature-map to obtain a cost volume. Then this cost volume passes through an optical flow estimator and a context network to produce the refined flow for a particular level.

We introduce the following changes to this architecture to obtain our modified (student) network GLOBALFLOWNET, as illustrated in Fig. 2 of [27]:

1. At each level, after the optical flow estimation, we add a LPM as described in Sec. 2.2. The cutoff frequency for the module is progressively made to increase from the coarse level to the fine level, up to a maximum of 8 (in both directions).

2. We also switch off the warping layer from the original PWC-Net, and instead use the motion estimated from the previous layer as an initial condition for the next one. We empirically observed better results as compared to using the warping layer. Moreover, [25, Table 5e] shows that excluding this layer does not adversely affect the optical flow accuracy significantly.

Note that at the time of deployment, only the student network GLOBALFLOWNET from Fig. 1 needs to be used. The teacher network plays a role only during training.

3. Video Motion Stabilization

Once GLOBALFLOWNET is trained, we use the optical flows produced by it to stabilize a video through a two-stage process. The first stage is a global motion stabilization stage involving correcting for the affine distortions in a novel way, detailed below in Sec. 3.1. This stage corrects a significant amount of global instability. However, since affine transformation is not a good representation for the global motion, this stage leaves behind some spatio-temporal distortions in the video (see also Sec. 4 and many video results in [27]). We correct these residual motion instabilities through a second stage (see Sec. 3.2) involving smoothing of DCT coefficient representation for the remaining global inter-frame motion.

3.1. Stage 1: Global Motion Stabilization

In this stage, we approximate the dense global motion as a coarser affine transformation, and then smooth the sequence of frame-to-frame affine transformation parameters.

Estimating Affine Transformations: Affine transformations between consecutive video frames are commonly obtained using salient feature point matching and a robust estimator involving RANSAC. However, this approach is expensive and error-prone if many salient points are concentrated on moving objects or in small regions of the image. Instead, we adopt a novel approach for affine estimation: (1) employing GLOBALFLOWNET to determine the smooth global motion $\boldsymbol{f_S}$ as detailed in Sec. 2.3, and (2) estimating the $K \triangleq 4$ parameters (rotation angle r, translation t_x, t_y and logarithmic scale s) of the (partial) affine transformation directly from the global motion as described in supplementary material [27]. The computation of global motion $\boldsymbol{f_S}$ enables an efficient and fitting of affine motion parameters between adjacent frames without any expensive schemes like RANSAC.

Parameter Sequence Smoothing: Consider a parameter sequence $\{\boldsymbol{\alpha_i}\}_{i=1}^{T-1}$ of affine transformations between consecutive pairs of frames $\{(I_i, I_{i+1})\}_{i=1}^{T-1}$ in the unstable video. Here $\boldsymbol{\alpha_i}$ is the vector of K parameters (enlisted earlier) of the affine transformation from frame I_i to I_{i+1}, and T is the total number of frames of the unstable video. We need to smooth $\{\boldsymbol{\alpha_i}\}_{i=1}^{T-1}$ to yield a resulting (smoothed) parameter sequence $\{\boldsymbol{\beta_i}\}_{i=1}^{T-1}$, and apply the residual motion sequence $\{\boldsymbol{\gamma_i}\}_{i=1}^{T-1}$, where $\forall i, \boldsymbol{\gamma_i} \triangleq \boldsymbol{\beta_i} - \boldsymbol{\alpha_i}$, to the frames of the unstable video to perform stabilization. However, excessive smoothing of $\{\boldsymbol{\alpha_i}\}_{i=1}^{T-1}$ can lead to a huge loss of field of view. When a video frame is warped using parameters $\boldsymbol{\gamma_i}$ for the purpose of stabilization, some parts of the frame contents will fall outside the usual rectangular canvas and intensity values will remain undefined in some parts. The ratio of the area of the largest inscribed rectangle inside the valid parts of the warped frame to the area of the original rectangular frame is called the *crop ratio* and denoted by C_R. We want to ensure that C_R is no less than some user-specified limit κ. For this, we need to constrain the values $\boldsymbol{\gamma_i}$ using slack parameters $\{\xi_k\}_{k=1}^{K}$ in such a way that $\forall k \in [K], |\gamma_i(k)| \leq \xi_k$. The cost function we seek to minimize in order to find a smoothed parameter sequence $\{\boldsymbol{\beta_i}\}_{i=1}^{T-1}$ is given as follows:

$$C_2(\{\boldsymbol{\beta_i}\}_{i=1}^{T-1}) \triangleq \sum_{i=2}^{T-1} \|\boldsymbol{\beta_{i+1}} - 2\boldsymbol{\beta_i} + \boldsymbol{\beta_{i-1}}\|_2^2$$

$$\text{such that } \forall i \in [T-1], \forall k \in [K], |\beta_i(k) - \alpha_i(k)| \leq \xi_k. \quad (3)$$

As per cinematography principles, smoothness of the underlying motion parameter sequence (MPS) is a key feature of stabilized videos, which $C_2(.)$ promotes. Eqn. 3 represents a constrained quadratic programming problem for which fast solvers exist.

Choice of slack parameters: The crop ratio C_R for any frame I_i is a decreasing function of $\{|\gamma_i(k)|\}_{k=1}^{K}$. Searching for all $\{\xi_k\}_{k=1}^{K}$ to maintain $C_R \geq \kappa$ is an expensive operation. For simplicity, we express $\forall k \in [K], \xi_k = \lambda_k z$ where λ_k is set to be equal to the average local standard deviation of the values from $\{\alpha_i(k)\}_{i=1}^{T-1}$. Given this, we now use binary search to select the maximum value of the parameter $z \in [0, 1]$ so that C_R does not fall below κ. Note that this is a single parameter for the entire video. A set of sample path results obtained by solving Eqn. 3 using $\{\xi_k\}_{k=1}^{K}$ thus selected, are presented in Fig. 2.

The main differences between our technique and related MPS smoothing approaches from [6, 15] are that (1) we compute the affine parameters without point matching (see Sec. 3.1), and that (2) we tune $\{\xi_k\}_{k=1}^{K}$ keeping the crop ratio in mind. On the other hand, the work in [6] puts upper/lower bounds on the values of $\beta_i(k)$, which are less intuitive to specify. The method in [15, Eqn. 5] penalizes a weighted combination of the smoothness of $\{\boldsymbol{\beta_i}\}_{i=1}^{T-1}$ and its similarity to $\{\boldsymbol{\alpha_i}\}_{i=1}^{T-1}$, without considering C_R.

3.2. Stage 2: Residual Motion Stabilization

Consider frame I_i at time instant i in the unstable video. Let us define $\Omega(i; W_R) \triangleq [i - W_R, i + W_R]$ to be a temporal neighborhood of radius W_R around frame I_i. Let the estimates of the global motion from I_i to all frames in $\{I_j\}_{j\in\Omega(i;W_R)}$ as produced by GlobalFlowNet be denoted by $\{\boldsymbol{f_S}^{(i,j)}\}_{j\in\Omega(i;W_R)}$. The local sequence $\{\boldsymbol{f_S}^{(i,j)}\}_{j\in\Omega(i;W_R)}$ will be temporally smooth in a stable video, since by design it contains no contribution from independently moving objects (see Sec. 2.3). Therefore, in order to stabilize the given video, we do the following: (1) We extract the global motion from I_i to $\{I_j\}_{j\in\Omega(i;W_R)}$ using GlobalFlowNet from Sec. 2.3, and (2) Apply temporal smoothing filters to smooth the low-frequency DCT coefficients $\{\boldsymbol{\theta}^{(i,j)}\}_{j\in\Omega(i;W_R)}$ representing the global motion sequence $\{\boldsymbol{f_S}^{(i,j)}\}_{j\in\Omega(i;W_R)}$.

For (2), we could have followed the quadratic programming strategy from Sec. 3.1. However we observed that a bilateral filter [28] of the following form with temporal smoothing parameter $\sigma_t \triangleq W_R/3$ and range smoothing parameter $\sigma_p \triangleq 0.1$ (for intensity values in [0,1]) yielded us good results:

$$\widetilde{\boldsymbol{\theta}}^{(i)} = \frac{\sum_{j\in\Omega(i;W_R)} e^{-(i-j)^2/2\sigma_t^2} e^{-\|I_i(\boldsymbol{\Psi\theta}^{(i,j)})-I_j\|^2/2N\sigma_p^2} \boldsymbol{\theta}^{(i,j)}}{\sum_{j\in\Omega(i;W_R)} e^{-(i-j)^2/2\sigma_t^2} e^{-\|I_i(\boldsymbol{\Psi\theta}^{(i,j)})-I_j\|^2/2N\sigma_p^2}}, \quad (4)$$

where $I_i(\boldsymbol{\Psi\theta}^{(i,j)})$ denotes the image I_i warped by the motion vector field $\boldsymbol{\Psi\theta}^{(i,j)}$ towards image I_j. Given the sequence of smoothed DCT coefficients $\{\widetilde{\boldsymbol{\theta}}^{(i)}\}_{i=1}^{T-1}$, the corresponding smoothed global motion estimates $\{\boldsymbol{\Psi}\widetilde{\boldsymbol{\theta}}^{(i)}\}_{i=1}^{T-1}$, where $\boldsymbol{\Psi}$ represents the 2D-DCT, are used to warp the frames $\{\widetilde{I}_i\}_{i=1}^{T-1}$ to generate the stabilized video frame with suitable cropping/resizing. In practice, better results were observed by not smoothing the sequence of zero-frequency DCT coefficients (i.e. DC), but performing smoothing on DCT coefficients of other frequencies. This is because the DC coefficients represent translational motion, and hence that sequence is already smooth due to the procedure in Sec. 3.1.

Our strategy here is a generalization of the affine MPS smoothing technique from [19] to smoothing DCT coefficient sequences. However, it is easy to compose different frame-to-frame affine transformations by iterated matrix multiplication (or parameter addition) to compute the affine motion from frame I_i to its neighbor I_j. This is not possible using the DCT coefficient representation. Hence we *separately* estimate the motion from I_i to every member of $\{I_j\}_{j\in\Omega(i;W_R)}$ using GlobalFlowNet.

3.3. Summary of Approach and Discussion

The exact sequence of steps for implementing our video stabilization approach are summarized in Alg. 1.

We note that the smoothness of warp fields has been earlier exploited in [35] based on piece-wise linear approximations, and in [36] using the Discrete Fourier Transform (DFT). However, we have observed better compactness using the DCT, which is in line with basic principles of image and signal processing [5, Fig. 8.25, Sec. 8.2.8] – see also Fig.1 of [27]. More importantly, our approach does not require elaborate pre-processing based on pre-trained segmentation masks to identify the foreground, any inpainting of the optical flow in occluded regions or any additional post-processing step on the output of the algorithm, unlike the approaches in [35, 36]. Therefore, our approach is simpler to implement.

Our paper presents a novel approach to global motion estimation, including affine transformation estimation, which does not use point tracking. The affine transformation estimates in steps 2–4 of Alg. 1, bring about a fair degree of stabilization to the original video. However as will be

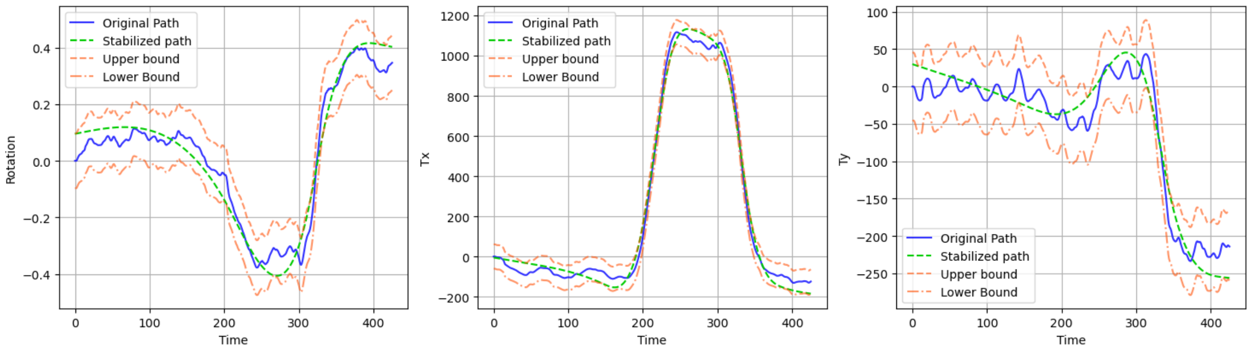

Figure 2: An illustration of affine MPS smoothing using the quadratic programming approach from Eqn. 3 for $\kappa = 0.8$ on a video from the 'quick rotation' category in [16].

shown later in Sec. 4 and in the accompanying video outputs in [27], they still retain a lot of wobble artifacts as well as geometric distortion. Due to this, the subsequent steps 5–7 for residual motion smoothing via DCT coefficients are also very important. The main reason for this is that the DCT-based global motion estimates form a compact but more general motion model than just 2D affine transformations. The approach in [17] also attempts to find global motion that is more general than 2D affine or homography transformations. However it adopts an iterative approach to detect moving objects and inpaint the flow in those regions, in conjunction with smoothing of the motion representation. Besides being iterative, their approach requires the selection of many parameters that may vary across iterations. Our approach here is much simpler to implement and does not require iterated feedback from the intermediate stabilized video.

```
ALGORITHM 1: Video Stabilization Algorithm
  Input: Input unstable video $\{I_i\}_{i=1}^{T}$; desirable crop ratio
         limit $\kappa$; smoothing window radius $W_R$.
  Output: Output stabilized video $\{J_i\}_{i=1}^{T}$.
1 Obtain the global motion estimates $\boldsymbol{f_S}^{(i,i+1)}$ from frame
    $I_i$ to $I_{i+1}$ for each $i \in [T-1]$ using the pre-trained
    GLOBALFLOWNET from Sec. 2.3.
2 For every $i \in [T-1]$, use a robust method to fit affine
    transformation parameters $\boldsymbol{\alpha_i}$ to $\boldsymbol{f_S}^{(i,i+1)}$ (Sec. 3.1).
3 Smooth the sequence $\{\boldsymbol{\alpha_i}\}_{i=1}^{T-1}$ to obtain the sequence
    $\{\boldsymbol{\beta_i}\}_{i=1}^{T-1}$ using the method in Sec. 3.1.
4 For each $i \in [T-1]$, warp the frame $I_i$ with suitable
    cropping to obtain an intermediate stabilized frame $\widetilde{I}_i$.
5 For every $i \in [T-1], j \in \Omega(i; W_R)$, determine estimates
    $\boldsymbol{f_S}^{(i,j)}$ of the global motion from $\widetilde{I}_i$ to $\widetilde{I}_j$.
6 For every $i \in [T-1], j \in \Omega(i; W_R)$, determine the DCT
    coefficients $\{\boldsymbol{\theta}^{(i,j)}\}$ of the global motion estimates from
    the earlier step. Smooth the temporal sequence of DCT
    coefficients using a window width $W_R$ in the method in
    Sec. 3.2 yielding $\{\widetilde{\boldsymbol{\theta}}^{(i)}\}_{i=1}^{T-1}$
7 Obtain the stabilized video by warping the frames
    $\{\widetilde{I}_i\}_{i=1}^{T-1}$ using the flow reconstructed from the smoothed
    DCT coefficients (with suitable cropping).
```

4. Experiments

Training Details: For global motion estimation, GLOBALFLOWNET was trained on randomly chosen consecutive frame pairs from 10K videos in the RealEstate10K dataset [38]. The pre-trained PWC-Net model from [25] was used for the teacher network. To train the student, we chose a batch-size of 16 and 200K iterations with the Adam optimizer. The training time was two days with NVIDIA Quadro RTX 5000 16GB GDDR6 Graphics Card and Intel Xeon E5-2620 CPU. Unlike [36], we do not perform stabilization, but only motion estimation, via a neural network. Hence no synthetic distortions need to be introduced in the dataset for training.

Dataset, Parameters and Comparisons: We now present experimental results to validate our approach on a total of 202 videos, i.e., on *all*(142) videos belonging to the following categories from the dataset [16]: regular, large parallax, quick rotation, crowd, running, zooming, as well all 60 videos from the Deepstab dataset [29]. For our algorithm, we set $W_R \triangleq 16$ in Sec. 3.2. Our algorithm was compared against the following recent state-of-the-art techniques: (1) the 'bundled camera paths' (BCP) approach from [15], (2) the 'learning video stabilization' (LVS) approach from [36] (which is shown to be faster and superior to the earlier approach in [35]), (3) the deep multi-grid warping (DMGW) approach from [30], (4) the neural network approach called PWStableNet from [37], and (5) the frame interpolation approach (DIFRNT) from [2]. For our approach, we compare the outputs from the affine-only stage (steps 1–4 from Alg. 1) to those from the complete algorithm. We term these as GlobalFlowNet-Affine and GlobalFlowNet-Full respectively. For all competing methods, we used the imple-

Dataset	Category	Original	BCP [15]	LVS [36]	DMGW [30]	DIFRNT [3]	PWStableNet [37]	GlobalFlowNet-Affine	GlobalFlowNet-Full
Stability ↑									
NUS	Regular	0.781	0.948	0.847	0.877	0.795	0.845	0.942	0.924
	Parallax	0.886	0.942	0.917	0.914	0.870	0.887	0.945	0.945
	QuickRotation	0.949	0.961	0.945	0.927	0.898	0.948	0.937	0.963
	Crowd	0.857	0.933	0.899	0.898	0.835	0.869	0.943	0.944
	Running	0.784	0.894	0.845	0.795	0.757	0.801	0.900	0.907
	Zooming	0.910	0.950	0.936	0.903	0.912	0.855	0.919	0.919
DeepStab	*NA*	0.757	0.965	0.878	0.823	0.811	0.819	0.926	0.928
ISI ↑									
NUS	Regular	0.617	0.892	0.870	0.812	0.971	0.766	0.880	0.914
	Parallax	0.680	0.797	0.802	0.702	0.975	0.742	0.816	0.837
	QuickRotation	0.692	0.821	0.841	0.715	0.976	0.769	0.785	0.806
	Crowd	0.710	0.852	0.848	0.674	0.972	0.790	0.849	0.864
	Running	0.580	0.781	0.809	0.656	0.974	0.690	0.775	0.802
	Zooming	0.656	0.819	0.818	0.715	0.973	0.736	0.793	0.822
DeepStab	*NA*	0.681	0.942	0.887	0.858	0.776	0.800	0.880	0.897
ITF ↑									
NUS	Regular	18.89	28.07	26.95	24.73	22.53	21.77	27.35	29.82
	Parallax	18.67	22.16	22.31	20.42	20.55	20.38	22.66	23.44
	QuickRotation	19.60	24.08	23.20	21.20	21.64	21.54	22.46	23.27
	Crowd	19.40	23.54	23.47	19.34	21.30	21.30	23.65	24.34
	Running	17.06	22.17	23.28	19.53	19.70	18.97	22.27	23.21
	Zooming	19.04	23.84	23.86	21.34	21.42	20.80	23.13	24.29
DeepStab	*NA*	20.173	25.362	26.885	26.171	22.608	21.951	27.099	28.195
Crop Ratio ↑									
NUS	Regular	1.000	0.834	0.865	0.567	0.969	-	0.843	0.827
	Parallax	1.000	0.867	0.780	0.438	0.970	-	0.838	0.814
	QuickRotation	1.000	0.845	0.555	0.453	0.658	-	0.835	0.808
	Crowd	1.000	0.853	0.825	0.483	0.962	-	0.831	0.802
	Running	1.000	0.826	0.701	0.476	0.942	-	0.799	0.767
	Zooming	1.000	0.768	0.711	0.513	0.903	-	0.834	0.791
DeepStab	*NA*	1.000	-	0.791	0.471	0.972	-	0.821	0.790
AGDMR ↑									
NUS	Regular	0.000	0.812	0.592	-0.384	0.305	0.411	0.735	0.821
	Parallax	0.000	0.439	0.312	-3.354	0.234	0.171	0.573	0.635
	QuickRotation	0.000	0.087	-2.360	-6.685	-2.069	-0.398	0.297	0.296
	Crowd	0.000	0.402	0.283	-3.518	0.265	0.232	0.567	0.632
	Running	0.000	0.687	0.598	-0.709	0.446	0.225	0.684	0.742
	Zooming	0.000	0.680	0.338	-1.165	0.160	0.178	0.625	0.686
DeepStab	*NA*	0.000	0.647	0.554	-0.019	0.278	0.486	0.786	0.815
Distortion ↑									
NUS	Regular	1.000	0.970	0.947	0.797	0.980	0.979	0.997	0.978
	Parallax	1.000	0.908	0.817	0.398	0.885	0.866	0.997	0.978
	QuickRotation	1.000	0.849	0.482	0.040	0.889	0.936	0.936	0.887
	Crowd	1.000	0.902	0.931	0.019	0.975	0.976	0.997	0.972
	Running	1.000	0.886	0.908	0.057	0.968	0.967	0.997	0.961
	Zooming	1.000	0.891	0.834	0.343	0.971	0.969	0.993	0.898
DeepStab	*NA*	0	0.647	0.554	-0.019	0.966	0.486	0.786	0.81

Table 1: Numerical performance comparison for different video stabilization methods. All measures are averaged across 142 videos from [16] and 60 videos from [29]. Higher values are better for all measures. *Note that the entries for Crop Ratios are left blank for methods without publicly available video results or the source code did not implement cropping invalid region*

mentations provided by the authors with their recommended parameters. For BCP, no code was available, but we compared with the sample results provided by the authors.

Visual comparison: Visual results for global motion estimation can be found in Fig. 1 and Figs. 2 and 3 of [27]. The visual outputs of the different methods can be observed in the supplemental material [27] on different videos from each of the six aforementioned categories. These results reveal the visual stability of GlobalFlowNet-Affine and GlobalFlowNet-Full, as compared to competing methods.

Numerical comparison: Subjective visual quality apart, we objectively compared the competing algorithms in terms of the following quality measures, results for which are presented in Table 1. 1. Stability [15]: This measure is computed by determining the frame-to-frame translation and rotation parameters in the stabilized videos. These parameter sequences (across time) are then reconstructed using only the first $K_F \triangleq 6$ DFT coefficients (other than DC). The stability measure is equal to the minimum (over the four parameters) of the ratios of the ℓ_2 norm of the reconstructed sequence to that of the original sequence. The intuition is based on the smoothness (dominant low frequency content) of these parameter sequences in stable videos. 2. Distortion [15]: This measure is computed from the average (across the frames) ratio of the smaller to the larger eigenvalue of the affine transformation matrix between the corresponding frames of the unstable and stabilized videos. A larger value (closer to 1) is desirable. This measure has some limitations as it will falsely yield an optimal value when the supposedly stabilized video is just a copy of the original unstable video (as the affine transformation between two identical frames is identity). We argue that other measures such as ISI, ITF and our new measure AGMDR introduced in the main paper, are more appropriate quality measures. But we are including comparisons using Distortion here, owing to its wide usage in video stabilization. 3. Inter-frame Similarity Index (ISI) [7]: This is the average inter-frame SSIM [31], expressed as $\frac{1}{T-1}\sum_{i=1}^{T-1}\text{SSIM}(I_i, I_{i+1})$. The intuition is that consecutive frames of a stabilized video would have higher pairwise SSIM than in an unstable one. 4. Inter-frame Transformation Fidelity (ITF) [7]: This is the average PSNR between consecutive frames and is expressed as $\frac{1}{T-1}\sum_{i=1}^{T-1}\text{PSNR}(I_i, I_{i+1})$. 5. Crop ratio [15] defined earlier in Sec. 3.1. 6. Average Global Motion Difference Ratio (AGMDR): This is a new measure which we propose here. It is equal to one minus the ratio of the total magnitude of the difference in the global motion between consecutive pairs of frames in the stabilized video to that in the unstable video. It is computed as $1 - \frac{\sum_{i=2}^{T-1}\|\boldsymbol{f_S^s}^{(i,i+1)} - \boldsymbol{f_S^s}^{(i-1,i)}\|_2}{\sum_{i=2}^{T-1}\|\boldsymbol{f_S^u}^{(i,i+1)} - \boldsymbol{f_S^u}^{(i-1,i)}\|_2}$. Here $\boldsymbol{f_S^s}, \boldsymbol{f_S^u}$ denote the inter-frame global motion in the stabilized and unstable videos respectively (unaffected by moving objects) and are computed using GLOBALFLOWNET from Sec. 2.2.

The values of all these measures except ITF lie between 0 to 1. Higher values for all these measures indicate better performance. The first three measures, though widely used in video stabilization [35, 36, 15, 14, 37], are based entirely on affine motion approximation, which as argued earlier, is an inaccurate motion model. On the other hand, AGMDR is based on a more general motion model. The intuition behind it is that the global motion across consecutive frames in a stable video should vary smoothly. Also, AGMDR is computationally very efficient, and by construction resilient against moving objects.

Discussion on results: Due to the absence of a single universally accepted evaluation measure for video stabilization, a holistic view of different needs/measures is required to draw a conclusion from numerical scores. On scores such as Stability, ISI, ITF and AGMDR, our techniques outperform the competing techniques LVS, BCP, DMGW and PWStableNet in most of the video categories, as can be seen in Table 1. Our methods outperform DIFRNT on all measures except ISI. However, we would like to note that, DIFRNT being based on iterative frame interpolation, would by design repeatedly reinforce similarity between adjacent frames over iterations, thus producing higher ISI. For AGMDR, some stabilization techniques produced jerkier optical flow in a few frames as compared to the original unstable video, leading to negative values of this measure. Our method produces less geometric/visual distortion than other methods as also evidenced by better ISI, ITF and Distortion scores (also see [27]). Also, the video 'Ablation.mp4' in [27] clearly shows the advantages of GLOBALFLOWNET-FULL over GLOBALFLOWNET-AFFINE.

5. Conclusion

We have presented a novel and intuitive video stabilization technique which uses a teacher-student network to obtain frame-to-frame global motion expressed by a small set of transform coefficients. We present novel ways to smooth the MPS in order to generate a stabilized video. Our method is quite general in nature and can potentially be extended using alternative architectures for optical flow (other than PWC-Net) and alternative transforms for motion representation (other than DCT). Our technique also inspired us to propose a novel quality measure to evaluate video stabilization. Moreover, the presented algorithm is computationally efficient and typically requires only ∼0.06 seconds and ∼0.5 seconds per frame (of size 480 × 640) for GlobalFlowNet-Affine and GlobalFlowNet-Full respectively.

The second stage GlobalFlowNet-Full of our algorithm will produce sub-optimal results if the area of a *single dominant* foreground object is large and comparable to that of the background. Note that in general, multiple independently moving small foregrounds do not pose a problem as their motion will generally be very different from each other and their combined motion will not dominate background) – see [27, Sec. 6]. In case of a single large foreground, the motion estimates will be biased towards either the foreground or the background, producing motion artifacts. The first stage GlobalFlowNet-Affine is more resilient to such situations. However a principled method to handle this in the second stage is left to future work. The cropping ratio can be further improved by adding [34]. The motion estimation can be improved using [9]. These are avenues for future work, as also extending our algorithm on rolling shutter videos and videos with significant motion blur.

References

[1] J. Barron. A general and adaptive robust loss function. In *CVPR*, 2019. 3

[2] J. Choi and I. S. Kweon. Deep iterative frame interpolation for full-frame video stabilization. *ACM TOG*, 39(1), 2020. 2, 6

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

[4] A. Goldstein and R. Fattal. Video stabilization using epipolar geometry. *ACM TOG*, 31(5), 2012. 2

[5] R. Gonzalez and R. Woods. *Digital Image Processing (Third edition)*. Pearson, 2013. 5

[6] M. Grundmann, V. Kwatra, and I. Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *CVPR*, 2011. 1, 3, 5

[7] W. Guilluy, A. Beghdadi, and L. Oudre. A performance evaluation framework for video stabilization methods. In *European Workshop on Visual Information Processing*, 2018. 8

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[9] Petr Hruby, Timothy Duff, Anton Leykin, and Tomas Pajdla. Learning to solve hard minimal problems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5532–5542, June 2022. 8

[10] A. Karargyris and A. Koulaouzidis. Odocapsule: Next-generation wireless capsule endoscopy with accurate lesion localization and video stabilization capabilities. *IEEE Transactions on Biomedical Engineering*, 62(1):352–360, 2015. 1

[11] A. Lim, B. Ramesh, Y. Yang, C. Xiang, Z. Gao, and F. Lin. Real-time optical flow-based video stabilization for unmanned aerial vehicles. *Journal of Real-Time Image Processing*, 16, 2019. 1

[12] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. Content-preserving warps for 3d video stabilization. *ACM TOG*, 28(3), 2009. 2

[13] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala. Subspace video stabilization. *ACM TOG*, 30(1), 2011. 2

[14] P. Liu, I. King, M. R. Lyu, and J. Xu. DDFlow: Learning optical flow with unlabeled data distillation. In *AAAI*, pages 8770–8777, 2019. 8

[15] S. Liu, L. Yuan, P. Tan, and J. Sun. Bundled camera paths for video stabilization. *ACM TOG*, 32(4), 2013. 1, 5, 6, 7, 8

[16] S. Liu, L. Yuan, P. Tan, and J. Sun. Video stabilization dataset. http://liushuaicheng.org/SIGGRAPH2013/database.html, 2013. 3, 6, 7

[17] S. Liu, L. Yuan, P. Tan, and J. Sun. Steadyflow: Spatially smooth optical flow for video stabilization. In *CVPR*, 2014. 1, 6

[18] Yu-Lun Liu, Wei-Sheng Lai, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. Hybrid neural fusion for full-frame video stabilization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2299–2308, October 2021. 2

[19] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum. Full-frame video stabilization with motion inpainting. *IEEE TPAMI*, 28(7), 2006. 1, 3, 5

[20] P. Milanovic, I. Popadic, and B. Kovacevic. Gyroscope-based video stabilization for electro-optical long-range surveillance systems. *Sensors*, 21(18), 2021. 1

[21] O. Oreifej, G. Shu, T. Pace, and M. Shah. A two-stage reconstruction approach for seeing through water. In *CVPR*, pages 1153–1160, 2011. 1

[22] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2

[23] Zhenmei Shi, Fuhao Shi, Wei-Sheng Lai, Chia-Kai Liang, and Yingyu Liang. Deep online fused video stabilization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1250–1258, January 2022. 1

[24] B. M. Smith, L. Zhang, H. Jin, and A. Agarwala. Light field video stabilization. In *ICCV*, 2009. 2

[25] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 3, 4, 6

[26] J. Sun. Video stabilization with a depth camera. In *CVPR*, 2012. 2

[27] Supplementary. Supplemental pdf accessible through cvf; source code and result videos at https://github.com/GlobalFlowNet/GlobalFlowNet, 2022. 4, 5, 6, 7, 8

[28] R. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV*, 1998. 5

[29] Miao Wang, Guo-Ye Yang, Jin-Kun Lin, Ariel Shamir, Song-Hai Zhang, Shao-Ping Lu, and Shi-Min Hu. Deep online video stabilization. https://github.com/cxjyxxme/deep-online-video-stabilization-deploy, 2018. 6, 7

[30] M. Wang, G.-Y. Yang, J.-K. Lin, S.-H. Zhang, A. Shamir, S.-P. Lu, and S.-M. Hu. Deep online video stabilization with multi-grid warping transformation learning. *IEEE TIP*, 2019. 2, 6, 7

[31] Z. Wang and A. C. Bovik. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009. 8

[32] J. Wulff and M. J. Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *CVPR*, 2015. 2

[33] S.-Z. Xu, J. Hu, M. Wang, T.-J. Mu, and S.-M. Hu. Deep video stabilization using adversial networks. *Computer Graphics Forum*, 2018. 2

[34] Yufei Xu, Jing Zhang, and Dacheng Tao. Out-of-boundary view synthesis towards full-frame video stabilization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4842–4851, October 2021. 2, 8

[35] J. Yu and R. Ramamoorthi. Robust video stabilization by optimization in CNN weight space. In *CVPR*, 2019. 2, 5, 6, 8

[36] J. Yu and R. Ramamoorthi. Learning video stabilization using optical flow. In *CVPR*, 2020. 2, 5, 6, 7, 8

[37] M. Zhao and Q. Ling. PWStableNet: Learning pixel-wise warping maps for video stabilization. *IEEE Transactions on Image Processing*, 29, 2020. 2, 6, 7, 8

[38] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 2018. 6

[39] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph.*, 2018. 2