Video Stabilization by Procrustes Analysis of Trajectories*

Geethu Miriam Jacob
Indian Institute of Technology
Chennai, India
geethumiriam@gmail.com

Sukhendu Das
Indian Institute of Technology
Chennai, India
sdas@iitm.ac.in

ABSTRACT

Video Stabilization algorithms are often necessary at the pre-processing stage for many applications in video analytics. The major challenges in video stabilization are the presence of jittery motion paths of a camera, large foreground moving objects with arbitrary motion and occlusions. In this paper, a simple, yet powerful video stabilization algorithm is proposed, by eliminating the trajectories with higher dynamism appearing due to jitter. A block-wise stabilization of the camera motion is performed, by analyzing the trajectories in Kendall's shape space. A 3-stage iterative process is proposed for each block of frames. At the first stage of the iterative process, the trajectories with relatively higher dynamism (estimated using optical flow) are eliminated. At the second stage, a Procrustes alignment is performed on the remaining trajectories and Frechet mean of the aligned trajectories is estimated. Finally, the Frechet mean is stabilized and a transformation of the stabilized Frechet mean to the original space (of the trajectories) yields the stabilized trajectories. A global optimization function has been designed for stabilization, thus minimizing wobbles and distortions in the frames. As the motion paths of the higher and lower dynamic regions become more distinct after stabilization, this iterative process helps in the identification of the stabilized background trajectories (with lower dynamism), which are used to warp the frames for rendering the stabilized frames. Experiments are done with varying levels of jitter introduced on stable videos, apart from a few benchmarked natural jittery videos. In cases, where synthetic jitter is fused on stable videos, an error norm comparing the groundtruth scores (scores of the stable videos) to the scores of the stabilized videos, is used for comparative study of performance. The results show the superiority of our proposed method over other state-of-the-art methods.

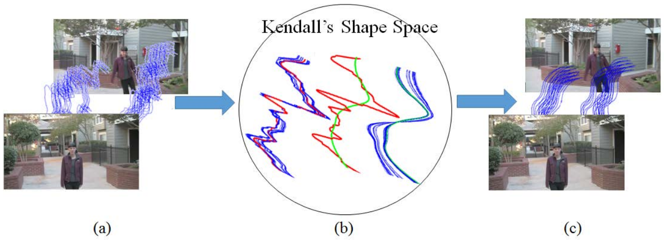

Figure 1: A schematic representation of the proposed video stabilization process. (a) Point trajectories from the regions with lower dynamism (background) are extracted across frames, (b) These are modelled in Kendall's shape space. The blue/red/green curves indicate the trajectories, the Frechet mean of the trajectories, the stabilized Frechet mean, respectively. (c) The stabilized Frechet mean is aligned to the original trajectory space to get the stabilized trajectories.

CCS Concepts

•Computing methodologies → Computer vision problems;

Keywords

Kendall's shape space, Procrustes Analysis, Frechet Mean, Video Stabilization, Content Preserving Warps

1. INTRODUCTION

With the advancement of the digital devices, videos taken by handheld cameras (e.g: portable camcorders and cellphones) are increasing day by day. Such videos are remarkably shaky (jittery) and unpleasant to the human eye. Video Stabilization is a technique of processing jittery videos captured using a non-stationary (e.g: handheld) camera to obtain smooth camera motion. Given a shaky video, the process of video stabilization renders a smooth video which is pleasant to the human eye. Many video stabilization softwares, like 'Youtube Stabilizer' and 'Warp Stabilizer' are available in the market to remove unwanted camera motion from these jittery videos. Inspite of some work done in this field of video stabilization, including commercial products, stabilizing videos with very high level of jitter, and stabilization of those videos with large moving foreground and occlusions, is still an open problem. When videos have moving objects and large depth variations, the motion estimation

*(Produces the permission block, and copyright information). For use with SIG-ALTERNATE.CLS. Supported by ACM.

ICVGIP, December 18-22, 2016, Guwahati, India
© 2016 ACM. ISBN 978-1-4503-4753-2/16/12. . . $15.00
DOI: http://dx.doi.org/10.1145/3009977.3009989

becomes challenging. When the trajectories belonging to the regions of moving objects are also used to estimate the camera motion as in [14] and [11], wobbles and distortions are generated. Thus, to reduce this effect, it is better to eliminate the trajectories belonging to the moving objects, regions near to the camera and strong depth edges (i.e regions with higher dynamism). In [18], the trajectories are optimized to be consistent with each other. This approach fails when there are many trajectories with higher dynamism. Also, in [5], user assistance is required for pruning the trajectories belonging to moving objects. Our method follows an iterative and automatic approach, where the trajectories with higher dynamism are eliminated in each iteration. As per our observation, stabilization increases the separability between trajectories with higher and lower dynamism.

In this paper, we propose a simple, yet powerful 2D video stabilization approach. The major steps in any video stabilization algorithm are motion estimation, motion smoothing and motion compensation. Our method focuses on the motion estimation and motion smoothing part. An illustration of the process is provided in Figure 1. We utilize point trajectories [6] to estimate motion of the video. These trajectories are long, dense and robust. The method models the extracted trajectories in Kendall's shape space [9]. A Procrustes analysis [9] on the trajectories is performed such that the trajectories are aligned to each other and smoothing is performed on the mean of the aligned trajectories, popularly known as the Frechet mean [9]. The stabilized Frechet mean can be utilized to obtain the stabilized trajectories. The advantage of this method is that a global stabilization is applied to every trajectory. Our method differs from other popular methods in that it stabilizes only the Frechet mean of the trajectories, whereas other methods perform stabilization or smoothing of every trajectory.

The contributions of this work are: 1) modelling the trajectories in Kendall's shape space so as to estimate the motion of the camera, 2) application of a global stabilization on the Frechet mean of the trajectories with lower dynamism instead of individually smoothing every trajectory, causing a significant reduction in wobbles and distortions and 3) experimentation on different levels of jitter applied to videos apart from the experiments on natural jittery videos, thereby analyzing the scalability of the method. The superiority of the proposed method is shown over two types of datasets: natural jittery videos and videos incorporated with different levels of synthetic jitter.

The rest of the paper is organized as follows. Section 2 discusses about the related works, section 3 gives a detailed explanation about the proposed framework. Section 4 discusses about the experiments conducted and the results obtained. Finally, section 5 concludes the paper.

2. RELATED WORKS

Based on the motion model and techniques used for the solution of the problem, a brief review of related video stabilization techniques is presented below, in three separate categories.

3D methods: The 3D methods estimate the 3D camera motion for stabilization. Given a 3D shaky camera path, the task of the stabilization process is to generate the corresponding virtual smooth 3D path. The stabilized video is obtained by rendering the video along the virtual path as if the video was shot through this virtual path. This rendering is often termed as view synthesis. Beuhler *et al.* [7] proposed a 3D video stabilization technique which performs a projective reconstruction of the video sequence. The stabilized projection matrices are computed by a nonlinear optimization on the reprojection error of the structure points. The method proposed in [20] minimizes the acceleration of the feature points in terms of relative pose sequences. The above methods take a set of shaky images from multiple viewpoints captured by a camera array, as input, and produces a single stable video as output. Liu *et al.* [13] proposes a novel view synthesis known as the Content Preserving Warps (CPW). This method generates 3D camera paths using a structure-from-motion (SFM) system, automatically fits the camera path to a user-specified path and then generates a spatially-varying warp from each input frame to obtain the stabilized frame. The work described in [22] extends the CPW to reduce the error in warping in textureless regions. The method ensures that planar regions have same homography. Depth sensors are utilized in [16] to get the depth image along with the shaky video. The 3D camera path is estimated from the color and depth images, smoothed and warped using CPW. In our work, we use CPW for rendering the frames after stabilization of trajectories.

2D methods - filtering and curve fitting: 2D methods of video stabilization utilize a series of 2D transformations or trajectories to represent the camera motion. Low pass filtering is a common method for smoothing the trajectories or transformations so as to obtain the desired camera path. Matsushita *et al.* [19] performs Gaussian smoothing of a series of neighboring transformations to obtain the desired set of transformations of the stabilized video. Also, [8] generates the smooth path by fitting a poly-line on the camera path. Gleicher and Liu [10] broke the trajectories into smaller segments and transformed each frame such that the video follows cimematic conventions. All the above 2D models are invalid when there is much variation of depths in the scene. The work of Liu *et al.* [14] smoothed trajectories by a simple low-pass filtering, automatic polynomial path fitting, and interactive spline fitting of some basis trajectories of the subspace, extracted from the feature tracks. This method has been transfered to Adobe After Effects [1] as a video stabilization function named 'Warp Stabilizer'. Recently, Liu *et al.* [15] extended the subspace method to deal with stereoscopic videos.

2D methods - variational methods: Several works pose the problem of 2D video stabilization as an optimization function. The method proposed by Grundmann *et al.* [11] obtains the desired camera path by posing a Linear Programming Problem (LPP). The LPP satisfies three constraints: Inclusive constraint, Proximity Constraint and Saliency Constraint. This method is integrated in the video enhancement functionality of Youtube. The method proposed in [21] models the trajectories as Bezier curves and performs a spatio-temporal optimization that finds smooth feature trajectories. Liu *et al.* [18] proposes a *bundled path model*, which parametrizes multiple, spatially-variant camera paths (a series of homography matrices across frames) that can deal with parallax and rolling shutter effects. Also, the authors extend the work to adapt the optimization function using optical flow [17]. There are several issues associated with video stabilization which need to be considered. Occlusion, motion blur, rolling shutter effects are some of them. To address the occlusion issue, Lee *et al.* [12] se-

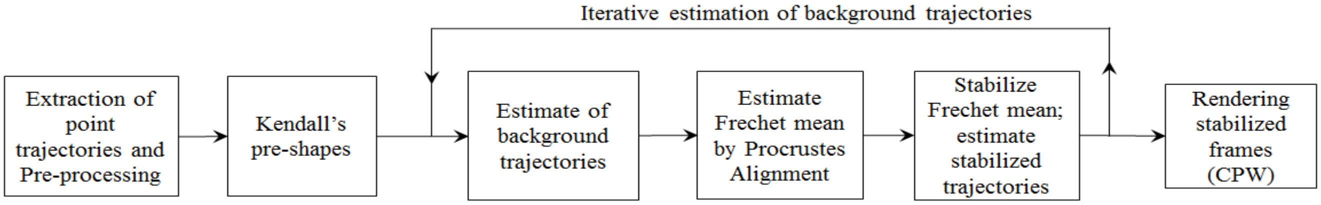
Iterative estimation of background trajectories
Extraction of point trajectories and Pre-processing
Kendall's pre-shapes
Estimate of background trajectories
Estimate Frechet mean by Procrustes Alignment
Stabilize Frechet mean; estimate stabilized trajectories
Rendering stabilized frames (CPW)

Figure 2: The detailed framework of the system for video stabilization for each block, by the Procrustes analysis of iteratively estimated background trajectories in Kendall's shape space.

lected robust feature trajectories and optimization of the set of transformations to smooth the trajectories. When the frames predominately contain the moving foreground object, all these methods fail [17]. The methods will not be able to estimate the camera motion of the background region because of the moving objects.

Here lies the motivation of our work to overcome these shortcomings. Our method is a 2D variational method, where we do not require the information on camera poses and depths as in many of the 3D methods. We aim at representing the camera motion by a series of transformations obtained from trajectories belonging only to the background regions. The stabilization is posed as an optimization problem, which reduces the variance of the trajectories across the frames. The novelty of the proposed method lies in the use of Kendall's shape space for trajectory representation, followed by Procrustes analysis for stabilization using an optimization function formulated specifically for this purpose.

3. PROPOSED FRAMEWORK

The proposed framework, as shown in figure 2, exhibits the stages of a novel method for performing video stabilization. The various stages of obtaining a smooth video from a shaky/jittery video are discussed below.

3.1 Block-wise partition of point trajectories

Given a jittery video as input, we extract point trajectories as introduced in [6] from a sequence of frames. Each point trajectory is represented by a sequence of coordinates. The entire video is divided into blocks based on the speed of camera movement. Observing the results on different categories of the videos analyzed, we have empirically obtained the block size. For videos with quick rotation or panning, we consider the number of frames in a block as 50. Otherwise, we take the size as 100. Also, the blocks overlap with a 10% extent of the size of the block. Only those trajectories which span the entire block are used for analysis. Consider the number of frames and the number of trajectories in a block to be N and K respectively. The k^{th} trajectory can be represented in the matrix form as:

$$X^k = \begin{bmatrix} x_1^k & x_2^k & \dots & x_N^k \\ y_1^k & y_2^k & \dots & y_N^k \end{bmatrix}^T$$

Here, X^k is the trajectory matrix, x_i^k and y_i^k represents the x and y coordinates of the k^{th} trajectory in i^{th} frame of the block.

The block-wise partition of frames ensure that enough point trajectories are obtained for the analysis, even though there are occlusions or fast movements of camera.

3.2 Trajectories as Kendall's pre-shapes

The trajectories are analyzed in Kendall's shape space [9]. A shape is defined as the geometrical information of a sequence or an object that is invariant under translation, scale and rotation transformations. The trajectory matrix of size $N \times 2$ is also known as a 'configuration' matrix. First, the centered pre-shape of the configuration is obtained. The centered pre-shape of the trajectory X^k is given as:

$$Z^k = \frac{C.X^k}{||C.X^k||_F}, \qquad (1)$$

where '.' is the matrix multiplication function, Z^k is the pre-shape of the k^{th} configuration and $C = I_N - \frac{1}{N} 1_N 1_N^T$. I_N is the $N \times N$ identity matrix and 1_N is the $N \times 1$ vector of ones. Pre-shape representation removes the location and scale information from the original configuration. A pre-shape space is the set of all possible pre-shapes. Thus, the trajectories are modelled in the pre-shape space, where they are centered with respect to the origin and are of unit length. The pre-shape space can be considered to be a unit hypersphere of 2(N-1) dimensions (for details and other properties, see [9]). The advantage of converting the trajectories to the pre-shape space is that, a plot of the coordinates of the pre-shapes gives a unified geometrical view of the shape of the original configurations (X^k) aligned with respect to a common frame of reference and location, the process of transformation, as given in Equation 1 for alignment, helps in comparing and analyzing different trajectories.

3.3 Estimate Background Trajectories

The trajectories that are extracted belong to different areas in the jittery video, namely moving objects, the areas belonging to the strong depth edges, and planar regions. It is observed that the trajectories belonging to the strong depth edges and the moving object have higher motion velocity, when compared to that of the background. Our aim is to estimate the jittery motion of the camera, which is assumed to provide trajectories moving with a lower motion velocity (or dynamism). We consider the trajectories with high motion velocity as trajectories with higher dynamism, and the others as those with lower dynamism (background trajectories). For a stable video, the motion of the trajectories in the regions with higher dynamism will be distinct and greater than that of the background. The moving objects will have a combined motion of both the camera as well as that for itself.

Identification of the background trajectories is done by an iterative process. Initially, a rough estimate of the background trajectories is done. Consider the accumulated motion vector of a trajectory k to be $c^k = \sum_{i=1}^{N} v_i^k$, where

v_i^k is the velocity vector of the trajectory k at frame i. In the first iteration, the velocity vectors of the trajectory is estimated, and in later iterations, velocity vectors of the stabilized trajectories are taken. An indicator function F defining whether a trajectory belongs to the background or otherwise is given as:

$$F_k = \begin{cases} 1, & \text{if } ||c^k - \frac{1}{K}\sum_{j=1}^{K} c^j||_2 < \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The indicator function value F_k, of the trajectory k is 1 when it belongs to the background and 0, otherwise. The indicator function identifies the trajectory as background if the the accumulated motion of the trajectory is much lower than the average accumulated motion of all trajectories. The threshold ϵ is adaptive and it varies with increasing iteration as $\epsilon = \frac{0.3}{n}$, where n is the iteration number.

The estimation of background trajectories is an iterative process, which is done after stabilization (described in Section 3.5) of the trajectories. It was observed in most of the cases that the system underperforms when $\epsilon < 0.1$, due to the lack of sufficient trajectories for motion estimation. The terminating condition for the number of iterations is when $\epsilon = 0.1$. Since we start with $\epsilon = 0.3$, we reach the termination condition in n=3 iterations (see Section 3.6 for more details and illustration).

3.4 Procrustes Alignment of Trajectories and Estimation of Frechet Mean

This module aims at estimating the shape of the mean of the background trajectories. A General Procrustes Analysis (GPA) [9] method is used for this purpose. GPA involves translating, rescaling and rotating the configurations relative to each other such that they align together. This is equivalent to minimizing a quantity proportional to the sum of squared norms of the pairwise differences of the configurations in the shape space. Modelling the original configurations in Kendall's pre-shape ensures that the translation and scale variations are removed. Let $\mathcal{K}$ be the number of background trajectories, i.e $\mathcal{K} = \sum_{i=1}^{K} F_i$. Optimum rotations required for aligning the configurations are estimated using the following optimization function:

$$\begin{aligned} \{\Gamma_i\} &= argmin_{\Gamma_i \in SO(2)} \frac{1}{\mathcal{K}} \sum_{i=1}^{K} \sum_{j=i+1}^{K} F_i F_j ||Z^i\Gamma_i - Z^j\Gamma_j||^2 \\ &= argmin_{\Gamma_i \in SO(2)} \sum_{i=1}^{K} F_i ||Z^i\Gamma_i - \frac{1}{\mathcal{K}} \sum_{j=1}^{K} F_j Z^j \Gamma_j||^2 \end{aligned} \tag{3}$$

where, the factor $[\frac{1}{\mathcal{K}} \sum_{j=1}^{K} F_j Z^j \Gamma_j]$ is the Procrustes estimate of the mean of the background trajectories, also known as the *Frechet mean* (μ), and $SO(2)$ is the Special Orthogonal group (of dimension 2), a subgroup of the Orthogonal group O(2) containing orthogonal matrices of determinant value 1. The solution to the function in Equation 3 is obtained using iterative least squares method. For more details, refer [9]. $\{\Gamma_i\}$ provides the rotation to be applied to the pre-shape configuration Z^i, to align itself to the mean shape μ. Any pre-shape Z^i is represented in shape space as $Z^i\Gamma_i$.

The Frechet mean is the mean of all the configurations in Kendall's shape space. Here, Frechet mean gives an estimate of the mean of the trajectories in the background. In other words, it is an estimate of the camera motion. We aim to reduce the unwanted shakiness in the camera motion globally, such that dynamic and non-dynamic regions are uniformly registered in the stabilized frame, thus reducing wobbles and distortions.

3.5 Stabilization using Frechet Mean of Trajectories

The Frechet mean acts as a representative trajectory of the camera motion. Hence, for obtaining stabilized trajectories in the shape space, stabilizing the Frechet mean would give the desired result.

For every block, we estimate the Frechet mean $\mu^m, m = 1, 2...M$, where M is the number of blocks. Also, each Frechet mean μ^m is stabilized using the optimization functions defined below in Equations 4 and 5. The stabilized Frechet mean $\hat{\mu}_s^1$ for the first block of frames is obtained using the following optimization function:

$$\hat{\mu}_s^1 = argmin_{\mu_s^1} (||\mu_s^1 - \mu^1||^2 + \lambda_1 \sum_{i=1}^{N} ||\mu_s^1(i,:) - \tilde{\mu}^1||^2) \tag{4}$$

The first term of the Equation 4 keeps the desired shape $\hat{\mu}_s^1$ similar to the input shape μ^1 and the second term minimizes the variance of the shape across the frames. For the rest of the blocks, the stabilized Frechet mean is obtained from the optimization function in Equation 5, given below

$$\begin{aligned} \hat{\mu}_s^m = argmin_{\mu_s^m} (||\mu_s^m - \mu^m||^2 + \lambda_1 \sum_{i=1}^{N} ||(\mu_s^m(i,:) - \tilde{\mu}^m)||^2 \\ + \frac{\lambda_2}{O} \sum_{j=1}^{O} ||(\mu_s^m(j,:) - \hat{\mu}_s^{m-1}(N - O + j,:))||^2) \end{aligned} \tag{5}$$

where, $\tilde{\mu}^m = \frac{1}{N} \sum_{j=1}^{N} \mu_s^m(j,:)$, O is the number of overlapping frames and λ_1 and λ_2 are two lagrangian multipliers which control the amount of smoothness of the trajectories and their consistency with the previous frame respectively. For Equation 5, a third term is added to the function in Equation 4, which ensures that the stabilized Frechet mean of the current block is consistent to the one in the overlapped frames of the previous block. This ensures that the stability is preserved across the block and there is no shakiness in the frames while rendering frames in the next block.

These optimization functions are solved using a jacobi iterative solver, by differentiating the optimization functions to provide a closed form solution. The closed form solution for the first block, as a solution to Equation 4, is as follows (t indicates the iteration index, taken as 20):

$$(\mu_s^1(r,:))^{t+1} = \alpha\mu^1(r,:) + \eta \sum_{i=1, i\neq r}^{N} (\mu_s^1(i,:))^t \tag{6}$$

where, $\alpha = (1 + \lambda_1 - \frac{\lambda_1}{N})^{-1}$ and $\eta = \frac{\lambda_1}{N}\alpha$
Similarly, the solution to the second function is

$$(\mu_s^m(r,:))^{t+1} = \gamma\mu^m(r,:) + \delta \sum_{i=1, i\neq r}^{N} (\mu_s^m(i,:))^t + \rho\hat{\mu}_s^{m-1}(i,:) \tag{7}$$

where, $\gamma = (1 + \lambda_1 - \frac{\lambda_1}{N} + \lambda_2)^{-1}$, $\delta = \frac{\lambda_1}{N}\gamma$ and $\rho = \frac{\lambda_2}{O}\gamma$ for the overlapping frames. For the non-overlapping frames,

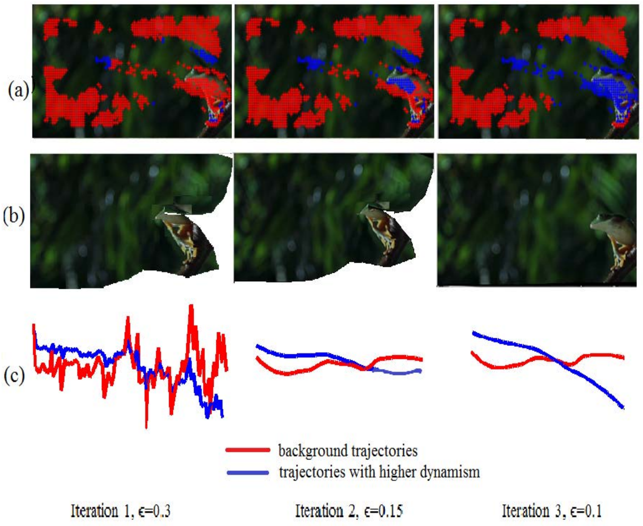

Figure 3: Iterative estimation of background trajectories. (a) Plot of coordinates of trajectories with higher dynamism (blue points) and lower dynamism (background trajectories, red points) for three iterations. (b) Output frame in each iteration, obtained with the background trajectories estimated in (a). (c) Trajectory plots of mean of background trajectories and that of trajectories with relatively higher dynamism. The means are mostly separable at the third iteration.

$\gamma = \alpha$, $\delta = \eta$ and $\rho = 0$. Thus at the end of the iterative process of applying Equations 4 and 5 to the Frechet mean, we obtain the stabilized Frechet mean $\hat{\mu}_s^m$, $\forall m$.

3.6 Stabilized Trajectories and Content Preserving Warps

To obtain the stabilized trajectories, the stabilized Frechet mean is transformed back to the original space of each trajectory. Let $\{Z^k\}$ be the set of configurations in block m. As obtained from Equation 3, for a pre-shape configuration Z^k, Γ_k represents the rotation matrix applied to align the pre-shape with the Frechet mean μ^m, i.e the Frechet mean can be transformed to the pre-shape as $Z^k = \mu^m \Gamma_k^T$. Similarly, the stabilized Frechet mean can be transformed to pre-shape of the trajectory, Z^k, to obtain its corresponding stabilized pre-shape $\tilde{Z}^k$, i.e $\tilde{Z}^k = \hat{\mu}_s^m \Gamma_k^T$. Later, the pre-shapes are transformed back to the original space and thus, we get a set of stabilized trajectories $\{\tilde{X}^k\}$ for each block of frames using the expression $\tilde{X}^k = \tilde{Z}^k(\frac{||C.X^k||_F}{C})$.

Given the original $\{X^k\}$ and stabilized trajectories $\{\tilde{X}^k\}$ and their coordinates, a spatially varying warp is applied to the frames, popularly known as "as-similar-as-possible" warping or Content Preserving Warps (CPW) [13]. The iterative process (see Figure 2) of estimating the background trajectories helps in the identification of regions of higher dynamism, thereby generating stable and wobble-free output frames. Figure 3 illustrates the process of iterative estimation of background trajectories. In the top row of the figure, the coordinates of the estimated trajectories with relatively higher dynamism (blue points) and those of background (red points) in a frame of the 'frog' video, from SegTrackV2 [2], are overlayed on the frames and shown for three iterations. All the trajectories belonging to the moving object (frog), having higher dynamism are identified as the background trajectories at the end of the third iteration. The middle row (Figure 3(b)) shows the output frames, obtained by analyzing the background trajectories estimated using Equation 2 for every iteration. Observe that the first and the second iterations generate frames with distortions/wobbles, as trajectories from regions with higher dynamism are also used for the analysis. The white spaces in the first two columns of Figure 3(b) appear due to unassigned pixels in the output frame, during the process of warping (CPW). This disappears at the third iteration (last column), when trajectories are stabilized. The third iteration generates the stabilized frames. The bottom row (Figure 3) illustrates the separability of the means of the background trajectories and those with relatively higher dynamism, both stabilized using Equations 4 and 5 with increasing iterations. At the left of the bottom row, the plot shows the means of the unstable trajectories, whereas the plots for second and third iterations, show the stabilized means of the trajectories estimated. The two means are most separable in the third iteration. Thus, at each iteration, the background trajectories are refined and correspondingly, the distortions are reduced. This is due to the fact that trajectories of the unstable video with higher and lower dynamisms becomes distinct after stabilization. The background trajectories estimated at the final iteration are used for further processing.

4. EXPERIMENTS AND RESULTS

The experiments were performed on two types of datasets: natural jittery videos (used in [18]) and synthetic jitter added to stable videos. Also, we extracted jitter from natural jittery videos and fused them on a few stable videos taken from SegTrackV2 dataset [2]. Different levels of jitter were added to the videos and experimented. The advantage of synthetic jitter fused into stable videos is that it helps in accurate performance analysis, as the groundtruth for stable videos can be extracted easily, which is not the case for natural jittery videos. Our method was compared with two commercial stabilizers, 'Youtube Stabilizer' and 'Warp Stabilizer' for the synthetic jitter based on the works reported in [11] and [14]. 'Youtube Stabilizer' is a parameter free online tool which allows the users to upload the video and download the stabilized video. 'Warp Stabilizer' is a tool that is interactive and allows the user to tune parameters. For the experimentation purpose, we used default parameter values. The results on natural jittery videos were also compared with the recent method [18] (results available from authors' website) along with the commercial stabilizers. As discussed in [18], the measures used for comparison are cropping, distortion or wobble and stability scores. The details on each score similarly, used in [18] for evaluating the performance of stabilization algorithms, are discussed below:

(a) Cropping score: This score is a measure of the cropping of frames while stabilizing. For natural jittery videos, the scale component of the homography between the input frame and the stabilized frame is extracted for this purpose. The average of the scale components over the frames is estimated. Ideally, the cropping score should be near to 1, indicating a small amout of cropping.

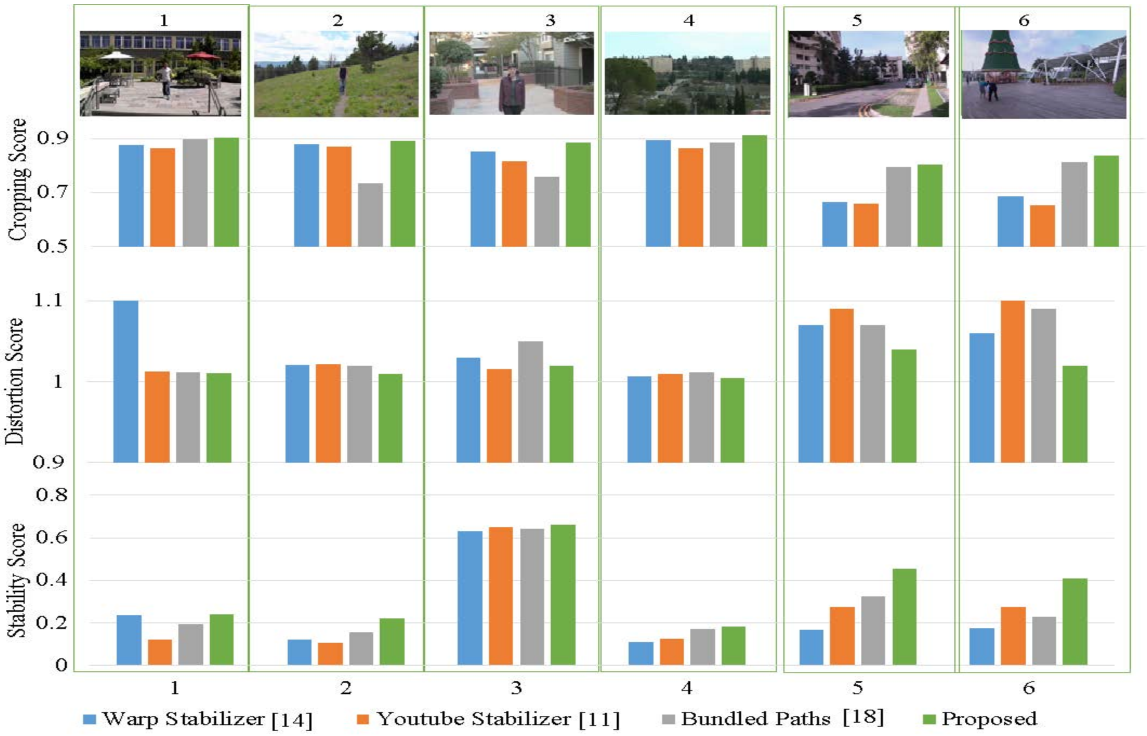

Figure 4: Performance analysis on video samples from the dataset [4]. The proposed method is compared with the state-of-the-art methods [11, 14] and [18]. The error measures are cropping score (ideally, near to 1), distortion score (lower the better, with a minimum of 1) and stability score (higher the better)

(b) Distortion/wobble score: The distortion score is indicated by the anisotropic scaling of the homographies between the input and output frames, which is computed as the ratio of the two eigenvalues of the affine part of the homography. Each frame has a wobble score, among which the worst one is taken as the final wobble score. Lower score indicates lower wobble with a minimum of 1 (no wobble).

(c) Stability score: To get the stability score, the feature tracks are analyzed in the frequency domain. The percentage of energy occupied by the low frequency components as a fraction of that over the entire frequency domain is calculated. Larger value of concentration on the lower frequency channels indicates higher stability.

For synthetic jittery videos, the groundtruth for stabilization is known (from that of stable videos available in the dataset). We measure the scores of the groundtruth and the stabilized output. An error norm is introduced to indicate how close is the stabilized output to the groundtruth.

4.1 Results on natural jittery videos

The experiments were done on 5 groups of challenging natural jittery videos from the dataset [4], namely simple (23 videos), zooming (23 videos), large parallax (20 videos), crowd (22 videos) and running (20 videos). Samples of six videos taken from this dataset are shown in the top row of Figure 4. The cropping, distortion and stability scores [18] of ours (proposed) and the state-of-the-art methods are shown as colored bar charts. 'Youtube Stabilizer' [11] and 'Warp Stabilizer' [14] are available for commercial use. We also compare the performance with the 'Bundled Paths' method [18], for which the results of the samples are available.

As seen in Figure 4, our results when compared with [11, 14, 18], for all the six samples, are better or comparable with the state-of-the-art methods. The first three videos have small amount of jitter, whereas the fourth video has zooming motion. The last two videos have large jittery motion of the camera and hence, the performances of all methods are worse when compared to that for the other videos. Our method outperforms the others in the cropping score. Concerning the distortion score, our method is comparable or better than all the other methods, i.e there is minimal distortion. Also, our method performs the best with the stability score as a metric (bottom row of Figure 4).

Figure 5 shows the failure cases (one frame per video shown) reported in works [14, 17], for which our method works well. The first video (Figure 5(a)) is a failure case of the method [14], where there is a prominent moving foreground. Our method eliminates the trajectories belonging to the moving foreground while performing Procrustes analysis, thereby minimizing wobbles and distortions. The second, third and fourth videos (Figure 5 (b), (c) and (d)) are failure cases of the method [17]. These videos have dominant foreground and our method again performs well for the videos (b) and (c) by considering only the background trajectories. For the case in Figure 5 (d), the proposed method performs well, but the stabilized video suffers from excessive cropping. The results can be downloaded from [3].

A few other cases [4] where our proposed method performs worse than the state-of-the-art are (Category (Id)): Large

Category	Scores	Warp Stabilizer [14]	Youtube Stabilizer [11]	Bundled Paths [18]	Proposed
Simple	Cropping Score	0.825	0.783	0.856	**0.871**
	Distortion Score	1.03	1.02	1.025	**1.01**
	Stability Score	0.349	0.305	0.447	**0.531**
Running	Cropping Score	0.643	0.715	0.810	**0.885**
	Distortion Score	1.32	1.73	1.13	**1.08**
	Stability Score	0.190	0.231	0.438	**0.517**
Crowd	Cropping Score	0.7734	0.774	**0.8556**	0.852
	Distortion Score	1.025	1.03	1.05	**1.02**
	Stability Score	0.3598	0.289	0.3601	**0.439**
Large parallax	Cropping Score	0.661	0.8361	**0.864**	0.854
	Distortion Score	1.029	1.039	1.02	**1.01**
	Stability Score	0.211	0.366	0.354	**0.374**
Zooming	Cropping Score	0.6067	0.8672	0.785	**0.877**
	Distortion Score	1.012	1.09	1.02	**1.011**
	Stability Score	0.156	0.346	0.432	**0.536**

Table 1: Comparison of the state-of-the-art methods [11, 14, 18] with the proposed method, using three score metrics averaged for five different categories of natural jittery videos.

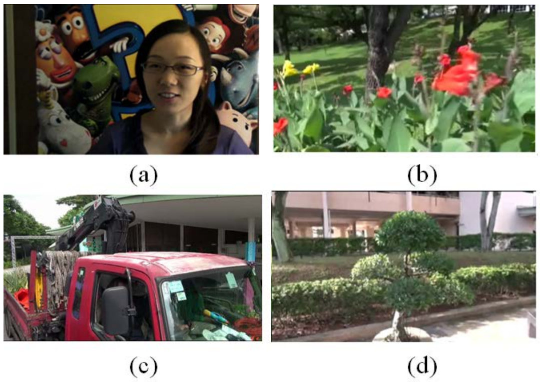

Figure 5: Failure cases of the methods (a) Subspace Stabilization [14], (b), (c) and (d) Steadyflow Stabilization [17]. Our method works for all these videos, except that (d) suffers from excessive cropping.

Parallex (4), Crowd (14), Zooming (21). The videos listed above have quick panning motion of the camera along with occlusions of objects, large number of moving objects and zooming along with quick panning of the camera. For such videos, the reasons for the unsatisfactory performance are:

(a) The point trajectories extracted are short and sparse, which produce wobbles in the stabilized video.

(b) The presence of a large number of moving objects lead to the inaccurate estimation of a Frechet mean. This in turn inaccurately represents the camera motion (Section 3.4), leading to distortions and wobbles in the stabilized frames.

The average scores for each category of natural jittery videos [4] are shown in Table 1. Different challenges are associated with each category. Videos belonging to the simple category are relatively less complex to handle, even though there are large depth variations. Videos belonging to crowd category produce occlusions; Running category has high amount of jitter with fast and varying camera motions; Parallax and zooming effects are produced due to manipulations and control of camera view, pose and motion. For the categories, 'running' and 'large parallex', the block size (parameter) is empirically set as 50, whereas for the rest of the categories, it is set as 100. Also, the default values of the lagrange multipliers (Equations 4 and 5) λ_1 and λ_2 are set as 0.5, except for the category 'running', where the values are set as 0.7 and 0.3 respectively. As seen in Table 1, the stability and distortion scores are the best for our proposed method in all the cases. The cropping score of the proposed method is better or comparable for all the cases. Thus, the results show that our proposed method stabilizes videos better under many different scenarios.

4.2 Results on synthetic jittery videos

As a part of evaluating the scalability of the methods with varying shakiness in the videos, different levels of jitter were fused into 3 stable videos. The stable videos were taken from the SegTrackV2 [2]. The jitter pattern was extracted from selected natural jittery videos [4]. Jitter synthesis is done by a relative comparison of the homographies of an unstable video to that of the corresponding stable video, stabilized using [17]. We assume that the homography of the unstable video is a product of the jitter component matrix and the homography of the stable matrix. This jitter matrix is extracted and warped with the input stable frames to get the jittery frames.

For each video, there are 3 levels of jitter added, namely JRL, JRM and JRH. JRL has lowest and JRH has the highest levels of jitter fused. For every frame, the rotation, shear and translation parameters of the jitter matrices are estimated and are perturbed by adding a random multiple (Gaussian distributed iid, with standard deviation σ) of the parameter values. The parameter σ controls the the level of jitter in the synthetic video. The values for σ are chosen as 0.05, 0.15 and 0.25 for the jitter levels JRL, JRM and JRH respectively. Considering the stable video itself as groundtruth, we estimate how close are the stabilized frames to the groundtruth. We measure the scores of the

Video	Jitter Level	Warp Stabilizer [14]	Youtube Stabilizer [11]	Proposed
Frog	JRL	1.335	1.311	**1.143**
	JRM	1.323	1.412	**1.058**
	JRH	1.349	1.765	**1.037**
Worm	JRL	1.567	1.287	**1.1205**
	JRM	1.581	1.455	**1.163**
	JRH	2.033	1.549	**1.112**

Table 2: Stabilization Error E (lower, the better) of the methods [11, 14] using the groundtruth (stable video available in the dataset).

input as well as that of the stabilized frames with respect to the groundtruth, for each method. The error measure for each method is calculated as follows:

Let $\beta_I = [1 - c_I, d_I, s_I]$ be the score vector of the input video, where c_I, d_I, s_I indicate the cropping, distortion and stability score of the input video with respect to the groundtruth (stable video). Now, let $\beta_O = [c_O, d_O, s_O]$ be the output score vector of the stabilized video. For a good stabilization algorithm, all three components of the absolute difference of the two vectors should be as large as possible. Then the error in the stabilization is given as $E = \frac{1}{||\beta_O - \beta_I||_2}$.

The metric E for the three methods are given in Table 2. For each level of the jitter, the error of each method with respect to the groundtruth is given. The videos [2] used are frog and worm. We have compared our method with 'Youtube Stabilizer' [11] and 'Warp Stabilizer' [14]. Our method performs the best for all the jitter levels, by a big margin in many cases.

The algorithm takes 4-5 seconds per frame (of size 640 x 360), when averaged over 10 videos, on an i7, 2GHz machine. Extraction of point trajectories and CPW take the majority share of the time taken, while the Procrustes analysis need negligible time.

5. CONCLUSION

An effective, simple and novel 2D method for video stabilization has been proposed. The background trajectories are modelled in Kendall's shape space, and a Procrustes alignment is performed to estimate the overall camera motion. Experiments were performed on natural jittery videos as well as jitter incorporated stable videos. The novelty of the proposed method lies in the formulation of an optimization function for stabilization, applied iteratively using the Frechet mean of the background trajectories. The method outperforms the state-of-the-art methods especially at high levels of jitter.

6. ACKNOWLEDGMENTS

This work is partially supported by Tata Consultancy Services (TCS).

7. REFERENCES

[1] Adobe After Effects. http://www.adobe.com/in/products/aftereffects.html.

[2] SegTrack v2 Dataset. http://web.engr.oregonstate.edu/~lif/SegTrack2/dataset.html.

[3] Stabilization Results. http://www.cse.iitm.ac.in/~vplab/stabResults.

[4] Video Dataset. http://liushuaicheng.org/SIGGRAPH2013/database.html/.

[5] J. Bai, A. Agarwala, M. Agrawala, and R. Ramamoorthi. User-assisted video stabilization. *Computer Graphics Forum*, 33(4):61–70, 2014.

[6] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2009.

[7] C. Buehler, M. Bosse, and L. McMillan. Non-metric image-based rendering for video stabilization. In *CVPR*, 2001.

[8] B.-Y. Chen, K.-Y. Lee, W.-T. Huang, and J.-S. Lin. Capturing intention-based full-frame video stabilization. *Computer Graphics Forum*, 27(7):1805–1814, 2008.

[9] I. L. Dryden. *Statistical shape analysis*, volume 4. John Wiley & Sons, 1998. ISBN:978-0-471-95816-1.

[10] M. L. Gleicher and F. Liu. Re-cinematography: Improving the camerawork of casual video. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 5(1):2, 2008.

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

[12] K.-Y. Lee, Y.-Y. Chuang, B.-Y. Chen, and M. Ouhyoung. Video stabilization using robust feature trajectories. In *ICCV*, 2009.

[13] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. Content-preserving warps for 3d video stabilization. *ACM Transactions on Graphics*, 28(3):44, 2009.

[14] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala. Subspace video stabilization. *ACM Transactions on Graphics*, 30(1):4, 2011.

[15] F. Liu, Y. Niu, and H. Jin. Joint subspace stabilization for stereoscopic video. In *ICCV*, 2013.

[16] S. Liu, Y. Wang, L. Yuan, J. Bu, P. Tan, and J. Sun. Video stabilization with a depth camera. In *CVPR*, 2012.

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

[18] S. Liu, L. Yuan, P. Tan, and J. Sun. Bundled camera paths for video stabilization. *ACM Transactions on Graphics*, 32(4):78, 2013.

[19] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum. Full-frame video stabilization with motion inpainting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1150–1163, 2006.

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

[21] Y.-S. Wang, F. Liu, P.-S. Hsu, and T.-Y. Lee. Spatially and temporally optimized video stabilization. *IEEE Transactions on Visualization and Computer Graphics*, 19(8):1354–1361, 2013.

[22] Z. Zhou, H. Jin, and Y. Ma. Plane-based content preserving warps for video stabilization. In *CVPR*, 2013.