

Moving Object Segmentation in Jittery Videos by Stabilizing Trajectories Modeled in Kendall's Shape Space

Geethu Miriam Jacob
geethumiriam@gmail.com

Sukhendu Das
<http://www.cse.iit.ac.in/~sdas/>

Visualization and Perception Lab
Dept. of CS&E
Indian Institute of Technology Madras
Chennai, India

Abstract

Moving Object Segmentation is a challenging task for jittery/wobbly videos. For jittery videos, the non-smooth camera motion makes discrimination between foreground objects and background layers hard to solve. While most recent works for moving video object segmentation fail in this scenario, our method generates an accurate segmentation of a single moving object. The proposed method performs a sparse segmentation, where frame-wise labels are assigned only to trajectory coordinates, followed by the pixel-wise labeling of frames. The sparse segmentation involving stabilization and clustering of trajectories in a 3-stage iterative process. At the 1st stage, the trajectories are clustered using pairwise Procrustes distance as a cue for creating an affinity matrix. The 2nd stage performs a block-wise Procrustes analysis of the trajectories and estimates Frechet means (in Kendall's shape space) of the clusters. The Frechet means represent the average trajectories of the motion clusters. An optimization function has been formulated to stabilize the Frechet means, yielding stabilized trajectories at the 3rd stage. The accuracy of the motion clusters are iteratively refined, producing distinct groups of stabilized trajectories. Next, the labels obtained from the sparse segmentation are propagated for pixel-wise labeling of the frames, using a GraphCut based energy formulation. Use of Procrustes analysis and energy minimization in Kendall's shape space for moving object segmentation in jittery videos, is the novelty of this work. Second contribution comes from experiments performed on a dataset formed of 20 real-world natural jittery videos, with manually annotated ground truth. Experiments are done with controlled levels of artificial jitter on videos of SegTrack2 dataset. Qualitative and quantitative results indicate the superiority of the proposed method.

1 Introduction

Moving Object Segmentation aims to segment a moving object, by classifying each pixel as belonging to either foreground or background. Moving Object Segmentation is a pre-processing step for high level applications such as video retrieval, video editing and activity recognition. Existing approaches segment the moving objects from either static videos [8, 20, 60, 82, 43] or moving camera videos [21, 28, 85, 66, 48]. However, very few works explore the performance of the segmentation algorithm in wobbly/jittery videos. Jittery videos are

taken from handheld devices, such as camcorders or cell phones, available ubiquitously with the users. Our method proposes to accurately segment a single moving object from jittery videos.

Moving object segmentation can be performed using unsupervised as well as semi-supervised methods. Trajectory based [9, 13, 19, 53, 55, 41] and graph based [14, 21, 28, 56, 47, 48] methods are popular in unsupervised techniques. Both the methods depends heavily on the accuracy of optical flow estimation and the distinctiveness of the motion between the moving objects and background. Hence, when the video is jittery, they fail due to the reduced distinctiveness in the motion between moving objects and background. Semi-supervised techniques [10, 22, 29, 57, 45] take user input in certain frames and the labels are propagated to the rest of the frames. This helps in tracking the moving object more precisely than the unsupervised techniques, even if the video is jittery in nature. But, when the video is long, much user input is required, which is undesirable.

We assume that there is a single moving object present in all the frames. Our method is an unsupervised trajectory based approach, which adopts the idea of stabilization to generate accurate clusters of trajectories. Given a jittery video as input, the method extracts the trajectories [44], stabilizes and clusters them separately as moving object and background trajectories. The stabilization aids in clustering the trajectories accurately. We propose a joint solution for the tasks of stabilization and segmentation, which needs less computational time than performing them separately. Also, the performance of the joint solution is better than executing the two tasks separately, since the quality of videos degrade after stabilization due to warping. We consider trajectories, as shapes and model them in Kendall's shape space [17] for stabilization and clustering, thus assigning labels to the trajectory coordinates in video frames (sparse segmentation). The sparse segmentation involving stabilization and clustering of trajectories in a 3-stage iterative process. First, the trajectories are clustered using pairwise Procrustes distance as a cue for creating an affinity matrix. Second stage performs a block-wise Procrustes analysis of the trajectories and estimates Frechet means (in Kendall's shape space) of the clusters. The Frechet means represent the average trajectories of the motion clusters. An optimization function has been formulated to stabilize the Frechet means, yielding stabilized trajectories at the third stage. The stabilized trajectories produce enriched clusters of motion paths for the next iteration. The accuracy of the motion clusters are iteratively refined, producing distinct groups of stabilized trajectories. Next, the labels obtained from the sparse segmentation are propagated for pixel-wise labeling of the frames, using a GraphCut based energy formulation.

The contributions of the work are as follows: 1) Procrustes distance as a similarity measure between trajectories for computing the affinity matrix for clustering, 2) Use of Procrustes analysis and a cost (energy) function for minimization in Kendall's shape space for the purpose of stabilization of trajectories, and 3) Developing a dataset of 20 real-world natural jittery videos with manual annotation of ground truth by us, for performance analysis. The superiority of the proposed method is shown on two datasets: a) 20 natural jittery videos and b) artificial jitter fused on videos of a standard (stable camera motion) segmentation dataset, SegTrack2 [4]. It is shown experimentally that the method works well for both stable as well as jittery videos. Our method generates accurate clusters in shape space, where many conventional trajectory clustering algorithms fail due to the randomness of the jittery motion. The better result of the proposed method is due to the stabilization of the trajectories, which makes the non-smooth foreground and background trajectories distinct.

The paper is organized as follows. Section 2 gives a brief review of work done on video stabilization and moving object segmentation. Section 3 describes the approach of trajectory

stabilization and clustering in Kendall’s shape space, whereas section 4 discusses about the procedure followed for pixel-wise labeling of frames from the sparse segments. Section 5 gives details of experiments and results obtained, while Section 6 concludes the paper.

2 Related Work

A brief review of various stabilization and segmentation methods are presented below:

Video Stabilization: Video stabilization is the process of smoothing the videos which contain considerable amount of wobble/jitter. Gaussian smoothing of neighboring transformations [60] and low pass filtering of trajectories in subspace [23, 24] are some of the methods developed for video stabilization. The works in [10, 16] generates the smooth path by fitting curves on the camera path. Meanwhile, several works pose the problem of video stabilization as an optimization function. The method proposed by Grundmann *et al.* [14] obtains the desired camera path by posing a Linear Programming Problem (LPP). Liu *et al.* proposed variational methods known as bundled paths model [26] based on camera paths and steadyflow model [27] based on optical flow. All the above methods perform smoothing or optimizations on all the trajectories or optical flow paths separately, due to which the stabilization may not be consistent across trajectories. The method proposed recently in [18] estimates a mean trajectory to represent and stabilize the camera motion. Our method is inspired from this work and aims at stabilizing and estimating the mean trajectories for both foreground and background motions.

Video Object Segmentation: The first kind of trajectory based segmentation method was introduced in [40], where motion profiles are segmented using normalized cuts [41]. Subspace clustering methods [13, 25, 68, 42] aims at clustering trajectories by modeling the trajectory matrix in subspaces. Brox *et al.* [9] proposed the concept of long term trajectories from optical flow. The long term trajectories track the pixels across frames. The clusters of the trajectories are converted to object segments using a variational model in [63]. The authors proposed an alternate clustering of long term trajectories using hypergraphs in [64]. Fragkiadaki *et al.* [15] cluster trajectories by computing discontinuities between trajectories that are Delaunay neighbors. The method in [10] uses global and local trajectory affinities to create clusters and later performs a GraphCut based energy minimization for generating segments. Potts’ model was used in [65] to generate segments of objects. Recently, Liu *et al.* [10] proposed a method to use both local and global affinities for spectral clustering, whereas the rest of the methods use local affinities.

Apart from trajectory based methods, many recent techniques propose novel unary and pairwise potentials for the GraphCut minimization. Lee *et al.* [21] ranks proposals in terms of appearance, objectness and motion and then perform spatio-temporal segmentation on the top ranked proposals. In [48], a layered directed acyclic graph based optimization is performed on the object proposals to obtain the segmented objects, whereas in [47], a saliency based proposal for unary potential is introduced. A two stage approach is proposed in [37], where the first stage produces an initial estimate of the object using inside-outside maps and the second stage refines the segmentation. A non local consensus voting on superpixels was performed in [42] to obtain the segmented objects. None of the above methods have been shown to work well for jittery videos. Our method is an unsupervised approach, which utilizes affinities of trajectories in Kendall’s shape space, defined using the Procrustes distance.

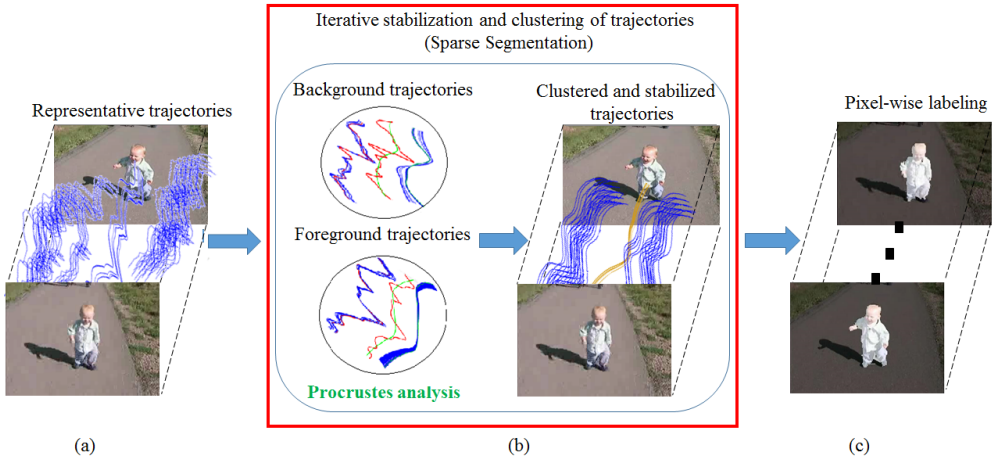


Figure 1: Proposed Framework: (a) Generate the representative trajectories and model them in Kendall’s shape space, (b) iterative stabilization and clustering using Procrustes Analysis of both foreground and background trajectories and (c) Pixel-wise labeling of frames.

3 Proposed Framework

The proposed framework is shown pictorially in Figure 1. The stages of our framework are:

1. *Model ‘representative trajectories’ in pre-shape space*: Point trajectories [24] are extracted and divided into blocks. For each block, representative trajectories are computed and modeled in Kendall’s pre-shape space.
2. *Iterative trajectory stabilization and clustering (sparse segmentation)*: The trajectories are stabilized and clustered iteratively using Procrustes analysis to obtain motion clusters. The stabilization of trajectories aids in producing well formed clusters.
3. *Pixel-wise labeling of frames*: The labels (of clusters) assigned to the trajectory coordinates are taken as input and a GraphCut based MRF formulation is applied to obtain the pixel-wise labeling. Appearance and location prior from superpixels of the clustered trajectory coordinates are taken as cues for the energy minimization.

3.1 Representative Trajectories as pre-shapes

Point trajectories are extracted using [24] and the frames are divided into varying-sized blocks based on the number of trajectories spanning the whole block. Empirically, the block size is selected such that the minimum number of trajectories spanning the entire block should be at least 10% of the total number of trajectories extracted. This ensures that the trajectories belonging to both foreground and background are included in the further processing. From the set of trajectories spanning an entire block, one trajectory is selected for each superpixel [5] in the first frame. Let N be the number of frames in the block and K be the number of representative trajectories selected. The k^{th} trajectory is represented in the

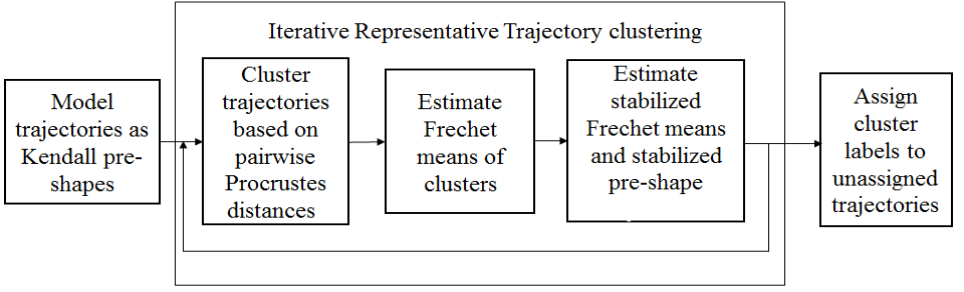


Figure 2: Iterative estimation of trajectory clusters: The process of clustering the trajectories, estimating the Fréchet means of the clusters and stabilizing them to obtain the stabilized trajectories are performed in an iterative manner. Each iteration refines the clusters. Later, unassigned trajectories are assigned cluster label based on their similarity with the estimated Fréchet means.

matrix form as:

$$X^k = \begin{bmatrix} x_1^k & x_2^k & \dots & x_N^k \\ y_1^k & y_2^k & \dots & y_N^k \end{bmatrix}^T \quad (1)$$

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.

As discussed in [14], a shape is defined as the geometrical information of a sequence of an object that is invariant under translation, scale and rotation transformations. A shape space is the set of all possible such shapes. Transforming the trajectory coordinates to the shape space ensures that they are aligned geometrically before any processing. First, the pre-shape configuration of the trajectory is computed. Pre-shape space, the space of all pre-shapes, is a precursor to the shape space, where the configurations are invariant to location and scale. The pre-shape of the trajectory X^k is given as:

$$Z^k = \frac{C.X^k}{\|C.X^k\|_F}, \quad (2)$$

where \cdot is the matrix multiplication function, Z^k is the pre-shape of the k^{th} configuration and $C = I_N - \frac{1}{N}1_N1_N^T$. I_N is the $N \times N$ identity matrix and 1_N is the $N \times 1$ vector of ones. The original trajectory configurations are centered and scaled to be of unit norm. Thus, the preshape space is a hypersphere of $2(N-1)$ dimensions (see [14] for more details).

3.2 Iterative Stabilization and Clustering of Trajectories (Sparse Segmentation)

The flowchart of the iterative procedure for stabilization and clustering of trajectories to obtain sparse motion segments is shown in Figure 2. Pairwise Procrustes distances between the representative trajectories are used to generate the affinity matrix for clustering, whereas Generalized Procrustes analysis (GPA) of the representative trajectories is performed to estimate and stabilize the motion of the foreground and background. Following are the three main stages for obtaining the clusters of trajectories:

3.2.1 Cluster trajectories using pairwise Procrustes distances

Pairwise Procrustes distances between the representative trajectories are considered for creating the affinity matrix for clustering. Trajectories belonging to the same cluster move together and hence should have higher affinity. Procrustes distance between two trajectories is defined as the distance between them in the shape space. The pre-shape configurations obtained from the previous stage are optimally rotated such that they become geometrically aligned. The transformation of the trajectories to the shape space makes them comparable. Let Z^i and Z^j be two trajectory configurations in the pre-shape space. The Procrustes distance between the pre-shapes is given as:

$$d_{proc}(i, j) = \|Z^i - Z^j \hat{\Gamma}\|_2 \quad (3)$$

where $\hat{\Gamma} = \operatorname{argmin}_{\Gamma \in SO(2)} \|Z^j - Z^i \Gamma\|_2$. Here $\hat{\Gamma} = UV^T$, where $Z^{jT} Z^i = V \Lambda U^T$ is the optimal rotation that aligns Z^i and Z^j , using the Singular Value Decomposition (SVD). This distance compares the shapes of the trajectories, which discursively tells the difference in the motion of the trajectories across the frames of the block. Lower the distance, the more similar their motion is.

The affinity matrix, which defines the affinity between the trajectories, for performing spectral clustering is created based on the pairwise Procrustes distance. The elements of the affinity matrix, Afy of size $K \times K$ is given as:

$$Afy(i, j) = \exp\left(-\frac{d_{proc}(i, j)}{\omega}\right) \quad (4)$$

The parameter ω is empirically estimated as 0.02 for optimal performance. Spectral clustering [46] is performed on the affinity matrix (Afy) created. This generates foreground and background clusters of trajectories. Thus, each trajectory is assigned a cluster label.

3.2.2 Estimate Frechet means of clusters

Generalized Procrustes analysis (GPA) as discussed in [47] is performed on each of the trajectory cluster. In GPA, the clusters are aligned in the shape space such that the estimates of the motion of the clusters is obtained. The pre-shape configurations are optimally rotated such that the sum of distances between the trajectories of the cluster in the shape space is minimized. Let the indicator function I_i^m for i^{th} pre-shape trajectory denote whether the trajectory belong to the m^{th} cluster or not and let \mathcal{K}^m be the number of trajectories in the m^{th} cluster, i.e $\mathcal{K}^m = \sum_{i=1}^K I_i^m$. Optimal rotations required for aligning the configurations are estimated using the following optimization function:

$$\begin{aligned} \{\hat{\Gamma}_i^m\} &= \operatorname{argmin}_{\Gamma_i^m \in SO(2)} \frac{1}{\mathcal{K}^m} \sum_{i=1}^{\mathcal{K}^m} \sum_{j=i+1}^{\mathcal{K}^m} I_i^m I_j^m \left[\|Z^i \Gamma_i^m - Z^j \Gamma_j^m\|_2 \right]^2 \forall m \\ &= \operatorname{argmin}_{\Gamma_i^m \in SO(2)} \frac{1}{\mathcal{K}^m} \sum_{i=1}^{\mathcal{K}^m} I_i^m \left[\|Z^i \Gamma_i^m - \frac{1}{\mathcal{K}^m} \sum_{j=1}^{\mathcal{K}^m} I_j^m Z^j \Gamma_j^m\|_2 \right]^2 \end{aligned} \quad (5)$$

where, the rotation matrices $\{\Gamma_i^m\}$ are constrained to belong to the Special Orthogonal group of dimension 2 (SO(2)), $\{\hat{\Gamma}_i^m\}$ are the optimal rotation matrices aligning the trajectories of each cluster together and the factor $[\frac{1}{\mathcal{K}^m} \sum_{j=1}^{\mathcal{K}^m} I_j^m Z^j \Gamma_j^m]$ is the Procrustes estimate of the mean

of the trajectories of m^{th} cluster. Since, this work focuses on the foreground background separation, the number of clusters (m) is set as 2.

The optimization function in Eqn 5 returns a set of transformations (rotation matrices, $\{\hat{\Gamma}_i^m\}, m = 1, 2$) for each cluster. The mean of the trajectories of the cluster in the shape space, popularly known as Frechet means, gives estimates of the motion of the clusters.

3.2.3 Estimate stabilized Frechet means and trajectories

Frechet means give the mean trajectories of the clusters in the shape space. Let the Frechet mean of the m^{th} cluster be given as $F_m = \frac{1}{\kappa^m} \sum_{j=1}^K I_j^m Z^j \hat{\Gamma}_j^m$. The Frechet means of the moving object and the background estimated are separately stabilized. The stabilization is performed using the optimization function:

$$\hat{F}_m^s = \underset{F_m^s}{\operatorname{argmin}} \left[\|F_m^s - F_m\|_2 \right]^2 + \lambda \sum_{i=1}^N \left[\|F_m^s(i, :) - \tilde{F}_m^s\|_2 \right]^2 \quad (6)$$

where λ is the tuning parameter (empirically set as 0.2 for low jitter and 0.6 for high jitter), F_m^s is the stabilized Frechet mean of the m^{th} cluster, \hat{F}_m^s is the optimal stabilized Frechet mean and $\tilde{F}_m^s = \frac{1}{N} \sum_{i=1}^N F_m^s(j, :)$. The first term in the function defined in Eqn 6 keeps the stabilized Frechet mean close to the original and the second term, a variance minimization term, reduces the jitteriness in the mean trajectory. The function is solved using a jacobi iterative solver, by differentiating the functions to obtain a closed form solution. The solution to the optimization problem in Eqn 6, is:

$$(\hat{F}_m^s(r, :))^{t+1} = \alpha F_m(r, :) + \beta \sum_{i=1, i \neq r}^N (\hat{F}_m^s(i, :))^t \quad (7)$$

where $\alpha = (1 + \lambda - \frac{\lambda}{N})^{-1}$ and $\beta = \frac{\lambda}{N} \alpha$. Here, t indicates the iteration index and is set to 5. Sparse segmentation is by itself an iterative process of alternating stabilization and clustering and hence number of iterations for the jacobi solver is set low. (Kindly refer the details given in App. A.1 in the supplementary file [9] for derivation of the closed form solution.)

To obtain the stabilized trajectories, the stabilized Frechet means are transformed back to the pre-shape space. For k^{th} trajectory, if $I_k^m = 1$, the stabilized pre-shape trajectory is obtained from the stabilized Frechet mean \hat{F}_m^s , as $Z^k = \hat{F}_m^s \Gamma_k^{mT}$. The three stages of sparse segmentation, namely clustering the shapes, Frechet mean estimation of the clusters and estimation of the stabilized pre-shape trajectories is an iterative procedure. Iteratively, the clusters become refined and accurate, and the Frechet means become more distinct. The number of iterations for obtaining the clusters is set as 3.

Once the iterative procedure ends, the trajectories which are not labeled and that span at least 70% of the frames in the block are assigned a cluster label. The trajectory is assigned the label of the mean which has the lowest Procrustes distance with itself. Our method is more robust to non-smooth trajectories, due to stabilization process. Also, the method is less sensitive to outliers. If outliers are present, the pixel-wise foreground labeling process, as described in the following section, refines and generates accurate segments of foreground and background.

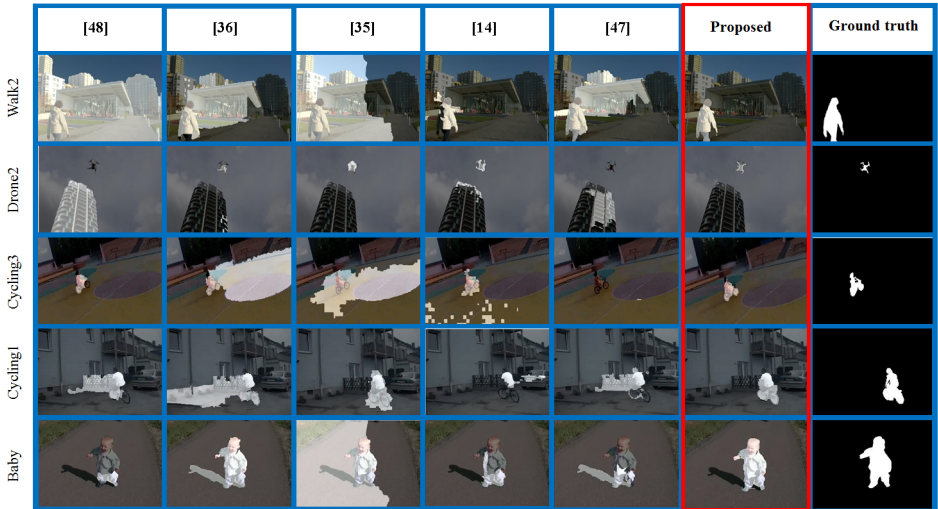


Figure 3: Visual comparison of the performance of our proposed method of tracking, with five state-of-the-art methods [14, 35, 36, 47, 48], shown with one frame each of the video shots: *Walk2*, *Drone2*, *Cycling3*, *Cycling1*, *Baby*. The column highlighted with the red template shows our result and the last column shows the ground truth of the frame.

3.3 Pixel-wise labeling of frames

Further, the superpixels [6] of the labeled trajectory coordinates in each frame are given the same label as that of the corresponding trajectories. MRF based optimization (as in [48]) is performed on the superpixels of the frame. Here, an appearance and location based unary potential, along with a color induced pairwise potential is defined. The optimization is performed using GraphCut algorithm [69]. This process classifies the superpixels in all frames as either foreground or background.

4 Experimental Results

Experiments were performed on two types of datasets: 20 real-world natural jittery videos; and a few videos from the standard segmentation dataset, SegTrack2, with artificial jitter fused in them to appear jittery. The proposed method is compared with 5 other state-of-the-art techniques [14, 35, 36, 47, 48], whose codes are publicly available. Only unsupervised methods are considered for the comparison with the proposed method. The measure used for performance evaluation is Intersection Over Union (IoU) [14, 27, 37], defined as the ratio of intersection to the union of the ground-truth and the segmented output.

A dataset of 20 real-world natural jittery videos with moving object was formed, among which 7 videos are from stabilization datasets [6] (*Walk1*, *Walk2*, *Cheery_Girl*, *Doll*, *Baby*, *Skating1*, *Car*), 5 jittery videos from HMDB51 dataset [9] (*Cycling1*, *Cycling2*, *Climb1*, *Climb2*, *Skating2*), 4 videos from the public resource, YouTube (*Drone1*, *Drone2*, *Drone3*, *Train*) and 4 videos shot by us (*Staircase1*, *Staircase2*, *Cycling3*, *Dog*) using a hand-held camcorder in motion. The ground truth of every fifth frame of each video was manually

Video samples	IoU scores of competing methods					Proposed
	[48]	[56]	[55]	[12]	[17]	
Walk2	0.009	0.123	0	0.480	0.151	0.841
Cheery_Girl	0.144	0.201	0.09	0.587	0.573	0.756
Doll	0.139	0.926	0.819	0.350	0.078	0.933
Baby	0.116	0.671	0.007	0.360	0.222	0.847
Skating1	0.033	0.248	0.318	0.627	0.523	0.713
Climb1	0.54	0.764	0.03	0.844	0.476	0.810
Drone2	0.487	0.436	0.549	0.325	0.348	0.588
Train	0.211	0.37	0.535	0.837	0.831	0.850
Cycling1	0.558	0.359	0.610	0.462	0.342	0.613
Staircase2	0.875	0.889	0.801	0.001	0.103	0.901
Average*	0.425	0.431	0.392	0.529	0.421	0.723
Single object?	Y	Y	N	N	Y	Y

Table 1: Performance analysis of segmentation on a subset of the 20 real-world jittery videos in terms of IoU score (higher, the better). See App. A.2 of the supplementary material for the individual results of all 20 videos [2]. Last row indicates whether the method handles single or multiple moving objects. * - Average over all real-world 20 videos.

annotated by us. The dataset and the ground truth can be downloaded from our website [4]. Figure 3 shows some qualitative results of all the five methods and ours on one frame on each of the videos: *Walk2*, *Drone2*, *Cycling3*, *Cycling1* and *Baby*. In all cases, unsatisfactory performance of prior work is visible. Our method performs the best for all the cases in the figure. Table 1 shows the quantitative results (on a subset of the 20 videos) for the experiments performed on natural jittery videos. As seen in the table, our method performs the best for all the videos except for one case, where ours is a very close second. On an average, as given in the second last row of table 1, our method performs the best by quite a margin. The last row indicates if the comparing method segments single moving object or multiple moving objects. The iterative method of stabilization using Procrustes analysis contributes to this enhanced performance. The method takes 5-6 seconds per frame (of size 360×640), when averaged over 10 videos, on an i7, 2GHz 16GB machine.

Jitter Level (σ)	IoU scores of competing methods					Proposed
	[48]	[56]	[55]	[12]	[17]	
Low (0.05)	0.586	0.637	0.327	0.692	0.535	0.695
Medium (0.15)	0.551	0.5749	0.525	0.686	0.506	0.690
High (0.25)	0.543	0.585	0.479	0.654	0.470	0.688

Table 2: Comparison of the average (over 72 videos) segmentation performances for low, medium and high levels of jitter, fused into stable videos (8) of SegTrack2 dataset [2].

Experiments were also done on videos with different artificial random jitter levels fused on video frames of SegTrack2 [2] dataset (8 videos chosen with a single moving object). A parameter σ (of Gaussian) controls the level of jitter fused on the stable videos to create 72 synthetic jittery videos. The values for σ are chosen as 0.05, 0.15 and 0.25 for low, medium and high levels of jitter. The same settings of perturbation is uniformly used for all

the methods. (App. A.3 of supplementary document [10] has details of the method used to fuse realistic random jitter patterns, using homography estimated across frames).

The segmentation performance of the proposed method on SegTrack2 [9], with different levels of jitter introduced, along with the state-of-the-art methods are shown in table 2. The average IoU metric over all the videos in the dataset for each level of jitter are shown. On an average, our method outperform others for all the three categories of jitter (Visual results of both natural and synthetic jitter videos are given in the website [11]).

To compare the performance of our method with the other state-of-the-art methods for stable videos, experiments were performed on four stable videos of Segtrack2 [9], for which quoted numbers from the papers of the competing methods are taken. Table 3 shows the comparison of four methods with the proposed method in stable videos. Average segmentation error (in pixels) measure is used to compare between the methods. Since the method [5] have not experimented on the videos from Segtrack2, the results are not quoted in the table. It can be seen that our method is comparable with the competing methods for all the four videos.

Video	Average segmentation error (in pixels) of competing methods				Proposed
	[18]	[36]	[12]	[47]	
Birdfall	155	239	146	209	150
Girl	1488	2404	797	1040	1012
Monkeydog	365	306	361	562	302
Parachute	220	347	219	207	199

Table 3: Comparison of average segmentation error for four videos of SegTrack2 dataset [9] quoting the reported numbers from the papers of the competing methods.

Limitations: Our method fails in the presence of independent uniform rigid motion (rotation or translation) throughout the frames. Such trajectories in Kendall’s shape space are difficult to differentiate (refer to the website [11] for observing the failure cases). Also, this method is experimented on a single moving object. Automatically identifying the number of motion segments and extending the proposed method to identify the independent motion segments with uniform rigid motion are the future directions of this work.

5 Conclusion

A novel trajectory clustering method for moving object segmentation in jittery videos based on analysis in Kendall’s shape space, using Procrustes distance as a cue, has been proposed in the paper. Trajectories are modelled in Kendall’s shape space and an iterative stabilization-clustering approach is followed for obtaining accurate foreground and background clusters. The stabilization of the trajectories is performed by Generalized Procrustes analysis (GPA) of trajectories. The stabilization improves the clustering accuracy and hence produces well formed clusters. Experiments were performed on 20 real-world natural jittery videos, for which the ground truth was manually annotated. Artificial jitter was fused on videos of a standard segmentation dataset. The performance analysis shows that the proposed method generates the best results when compared to all other state-of-the-art methods.

Acknowledgment- This work has been partially supported by TCS Foundation, India.

References

- [1] Jitter Video Segmentation in Kendall Shape Space . http://www.cse.iitm.ac.in/~vplab/BMVC_17/KendallSeg.html, .
- [2] Supplementary Material. http://www.cse.iitm.ac.in/~vplab/BMVC_17/KendallSegSupp.pdf, .
- [3] Serre Lab \hat{A} HMDB: a large human motion database. <http://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/>.
- [4] SegTrack v2 Dataset. <http://web.engr.oregonstate.edu/~lif/SegTrack2/dataset.html>.
- [5] Video Database. <http://liushuaicheng.org/SIGGRAPH2013/database.html/>.
- [6] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI*, 34(11):2274–2282, 2012.
- [7] S Avinash Ramakanth and R Venkatesh Babu. Seamseg: Video object segmentation using patch seams. In *CVPR*, 2014.
- [8] Olivier Barnich and Marc Van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *IEEE TIP*, 20(6):1709–1724, 2011.
- [9] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. *ECCV*, 2010.
- [10] Bing-Yu Chen, Ken-Yi Lee, Wei-Ting Huang, and Jong-Shan Lin. Capturing intention-based full-frame video stabilization. *Computer Graphics Forum*, 27(7):1805–1814, 2008.
- [11] Lin Chen, Jianbing Shen, Wenguan Wang, and Bingbing Ni. Video object segmentation via dense trajectories. *IEEE Transactions on Multimedia*, 17(12):2225–2234, 2015.
- [12] Mardia Dryden. *Statistical shape analysis*, volume 4. John Wiley & Sons, 1998. ISBN 978-0-471-95816-1.
- [13] Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *CVPR*, 2009.
- [14] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014.
- [15] Katerina Fragkiadaki, Geng Zhang, and Jianbo Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR*, 2012.
- [16] Michael L Gleicher and Feng Liu. Re-cinematography: Improving the camerawork of casual video. *ACM TOMM*, 5(1):2, 2008.
- [17] M. Grundmann, V. Kwatra, and I. Essa. Auto-directed video stabilization with robust 11 optimal camera paths. In *CVPR*, 2011.

- [18] Geethu Miriam Jacob and Sukhendu Das. Video stabilization by procrustes analysis of trajectories. In *ICVGIP*, 2016.
- [19] Suyog Dutt Jain and Kristen Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*. Springer, 2014.
- [20] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth extraction from video using non-parametric sampling. In *ECCV, 2012*. Springer, 2012.
- [21] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *ICCV*, 2011.
- [22] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013.
- [23] Feng Liu, Michael Gleicher, Jue Wang, Hailin Jin, and Aseem Agarwala. Subspace video stabilization. *ACM TOG*, 30(1):4, 2011.
- [24] Feng Liu, Yuzhen Niu, and Hailin Jin. Joint subspace stabilization for stereoscopic video. In *ICCV*, 2013.
- [25] Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *ICML*, 2010.
- [26] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM TOG*, 32(4):78, 2013.
- [27] Shuaicheng Liu, Lu Yuan, Ping Tan, and Sun Jian. Steadyflow: Spatially smooth optical flow for video stabilization. In *CVPR*, 2014.
- [28] Tianyang Ma and Longin Jan Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, 2012.
- [29] Nicolas Märki, Federico Perazzi, Oliver Wang, and Alexander Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, 2016.
- [30] Yasuyuki Matsushita, Eyal Ofek, Weina Ge, Xiaoou Tang, and Heung-Yeung Shum. Full-frame video stabilization with motion inpainting. *IEEE TPAMI*, 28(7):1150–1163, 2006.
- [31] Anurag Mittal and Nikos Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *CVPR*, 2004.
- [32] Anurag Mittal, Antoine Monnet, and Nikos Paragios. Scene modeling and change detection in dynamic scenes: A subspace approach. *CVIU*, 113(1):63–79, 2009.
- [33] Peter Ochs and Thomas Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, 2011.
- [34] Peter Ochs and Thomas Brox. Higher order motion models and spectral clustering. In *CVPR*. IEEE, 2012.
- [35] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *IEEE TPAMI*, 36(6):1187–1200, 2014.

- [36] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013.
- [37] Federico Perazzi, Oliver Wang, Markus Gross, and Alexander Sorkine-Hornung. Fully connected object proposals for video segmentation. In *ICCV*, 2015.
- [38] Shankar R Rao, Roberto Tron, René Vidal, and Yi Ma. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In *CVPR*, 2008.
- [39] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM TOG*, 23(3):309–314, 2004.
- [40] Jianbo Shi and Jitendra Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, 1998.
- [41] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8):888–905, 2000.
- [42] Mahdi Soltanolkotabi, Ehsan Elhamifar, Emmanuel J Candes, et al. Robust subspace clustering. *The Annals of Statistics*, 42(2):669–699, 2014.
- [43] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *CVPR, 1999*. IEEE, 1999.
- [44] Narayanan Sundaram, Thomas Brox, and Kurt Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*. 2010.
- [45] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J Black. Video segmentation via object flow. In *CVPR*, 2016.
- [46] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4): 395–416, 2007.
- [47] Wenguan Wang, Jianbing Shen, and Fatih Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, 2015.
- [48] Dong Zhang, Omar Javed, and Mubarak Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, 2013.