Robust Background Identification for Dynamic Video Editing

Fang-Lue Zhang[1] Xian Wu[1] Hao-Tian Zhang[1] Jue Wang[2] Shi-Min Hu[1,3]
[1] Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing
[2] Adobe Research [3] Cardiff University

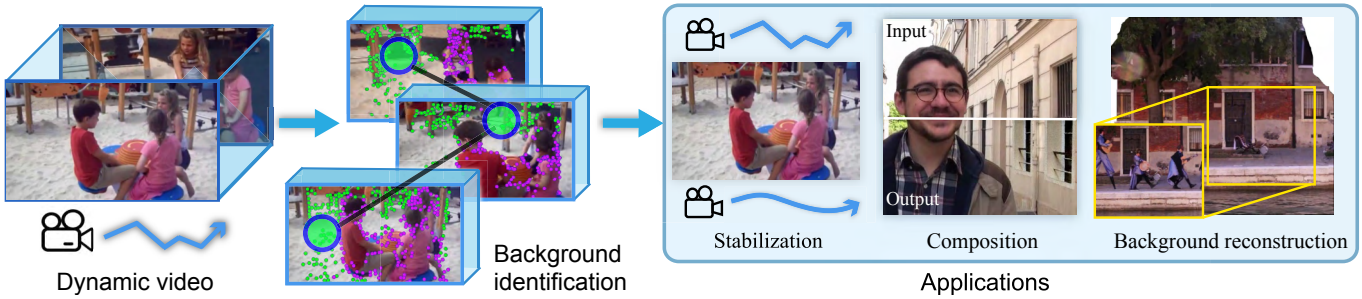

Figure 1: *We address the problem of background identification in highly dynamic videos, i.e., reliably classify background features from all those extracted from video frames. As an essential step for robust camera motion estimation, our method directly leads to significant improvements in applications such as stabilization, video composition and background reconstruction.*

Abstract

Extracting background features for estimating the camera path is a key step in many video editing and enhancement applications. Existing approaches often fail on highly dynamic videos that are shot by moving cameras and contain severe foreground occlusion. Based on existing theories, we present a new, practical method that can reliably identify background features in complex video, leading to accurate camera path estimation and background layering. Our approach contains a local motion analysis step and a global optimization step. We first divide the input video into overlapping temporal windows, and extract local motion clusters in each window. We form a directed graph from these local clusters, and identify background ones by finding a minimal path through the graph using optimization. We show that our method significantly outperforms other alternatives, and can be directly used to improve common video editing applications such as stabilization, compositing and background reconstruction.

Keywords: Feature point trajectory, background detection, video enhancement, video stabilization, camera path estimation

Concepts: •**Computing methodologies** → **Image manipulation;** *Computational photography;*

1 Introduction

Estimating camera motion from a video sequence is a fundamental task for many video editing and enhancement applications. For instance, videos captured by hand-held cameras often have unsteady and un-directed camera motion, making them unpleasant to watch. Video stabilization aims at solving this problem [Grundmann et al. 2011; Liu et al. 2013b], and extracting sparse feature points for estimating the original camera motion is typically the first essential step in existing approaches. It is also a necessary step for efficient edit propagation in video. For instance, for the task of inserting a new object in the background of a video shot by a moving camera, if the camera motion can be reliably estimated, the user can simply place the object in the first frame, and have it propagated automatically to the rest of the sequence.

In previous work, camera motion is usually estimated by extracting sparse feature points from each video frame, and matching them across frames for computing inter-frame transformations such as homographies [Grundmann et al. 2011; Zhang et al. 2015]. In other works, the camera motion is implicitly represented by a collection of long-range feature trajectories [Rao et al. 2010; Liu et al.]. A common assumption made in previous work is that the extracted feature points, or the vast majority of them, especially those that can be tracked for a long range, are located in the static background regions of the video. Their displacements across frames are only caused by camera motion. For improved robustness, RANSAC is often used to filter out outliers from the tracked features. This simple feature selection strategy however is insufficient to handle dynamic videos that contain large moving objects (see Fig. 3). In such videos, since the background is heavily occluded, the majority of features one can extract are instead located on the moving objects. Furthermore, given the intense object and camera motion, the revealed portion of the background is constantly changing, making long range background tracking impossible. Due to the lack of a more robust feature selection mechanism, existing video editing approaches often cannot reliably estimate the original camera motion in such cases, and report it as a typical failure mode that is well documented in the literature [Liu et al. ; Liu et al. 2013b].

In this work, we present a new *Background Identification* method that can reliably identify background features in videos of highly dynamic scenes. Our work is grounded on the well-known theory in computer vision that point trajectories belonging to different motions can be treated as living in different linear subspace of dimension 4 or less [Tomasi and Kanade 1992; Boult and Brown 1991]. It serves as the foundation for various motion matrix decomposi-

SA '16 Technical Papers,, December 05-08, 2016, Macao
ISBN: 978-1-4503-4514-9/16/12...$15.00
DOI: http://dx.doi.org/10.1145/2980179.2980243

ACM Reference Format
Zhang, F., Wu, X., Zhang, H., Wang, J., Hu, S. 2016. Robust Background Identification for Dynamic Video Editing. ACM Trans. Graph. 35, 6, Article 197 (November 2016), 12 pages. DOI = 10.1145/2980179.2980243 http://doi.acm.org/10.1145/2980179.2980243.

tion strategies [Costeira and Kanade 1998; Wu et al. 2001; Yan and Pollefeys 2005] for motion segmentation or in background motion subtraction [Elgammal et al. 2000; Cui et al. 2012]. However, as we will discuss more in the next section, previous background subtraction and motion analysis methods make strong assumptions on the input video such as static camera; the existence of a large number of long range feature trajectories; moving objects have to be few and small, etc. These assumptions often do not hold in real-world videos, especially amateur ones. In contrast, our approach makes weaker assumptions that are valid in more cases: (1) *partial visibility:* we assume a portion of the background is always visible in the video, although its size could vary and be small; (2) *short-range trackability:* we assume in any short period of time (e.g., one second), a number of background features can be extracted and tracked, although they could be far less than the foreground features in the same period. As we will show later, these assumptions are often valid even in very challenging examples (see Fig. 5).

Following these assumptions, we propose a two-level approach for background identification. In the *local motion analysis* step, we divide the video into overlapping local temporal windows: in each we cluster features into local feature groups as background hypotheses. At the second, *global optimization* step, treating each hypothesis as a graph node, we perform a spatio-temporal graph optimization to choose local feature groups that together can yield coherent camera motion for the entire video. As an advanced feature selection tool, our method directly leads to better, more robust camera motion estimation, thus can greatly improve many existing video editing tasks in difficult cases. As examples, we show that our method can be applied for important tasks including better video stabilization, background reconstruction and video object composition.

2 Related Work

We now briefly review most related works.

Dense segmentation and background extraction Segmenting video frames into semantic regions is a fundamental problem in computer vision [Fragkiadaki and Shi 2011; Perazzi et al. 2015; Taylor et al. 2015]. There are also methods on video over-segmentation to provide more compact video descriptions for high-level applications such as object extraction [Van den Bergh et al. 2012; Chang et al. 2013; Zhang et al.]. Background subtraction, i.e., constructing a clean background plate and using it to extract moving objects in video, is one of the segmentation tasks that is most related to our work.

Most background extraction methods focus on reconstructing the background for static surveillance cameras [Elgammal et al. 2000; Barnich and Van Droogenbroeck 2009; Cheng et al. 2011], hence cannot handle moving cameras. To relax this restriction, Hayman and Eklundh [Hayman and Eklundh 2003] used homography transformations for frame alignment and identify regions that are consistent with the transformations as background. This approach fails when large moving objects present and homographies cannot be estimated accurately, a problem we try to solve in this work. Zhang et al. [2007] use depth information for more robust foreground segmentation. However, accurate depth itself is hard to extract for complex scenes with multiple moving objects. More recently, Chiu et al. [2010] and Mumtaz et al. [2014] proposed new probabilistic models to solve this problem, under the assumption that the background region is almost the same in the entire video. It thus cannot work with videos captured with more intense camera movement, such as panning the camera to sweep through a large scene. Papazoglou et al. [2013] proposed a robust object extraction method that performed well on videos captured by fast-moving cameras, but it is limited to allow only one moving object.

Motion segmentation on feature trajectories Grouping feature trajectories based on motion information is highly related to our work. In previous methods, feature point trajectories are first extracted by feature tracking, and then arranged together to form a large matrix. Matrix decomposition method [Tuzel et al. 2005; Vidal et al. 2008] and compressed-sensing-based data coding methods [Ma et al. 2007] are used to find the best clustering of the trajectories. Rao et al. [2010] proposed a method to deal with incomplete, or corrupted trajectories. Luo et al. [Luo and Huang 2014] introduced an adaptive manifold model to describe the trajectories. For long-range trajectories, Sand et al.[2008] and Brox et al. [2010a] proposed robust motion estimation and segmentation methods. These approaches do not explicitly identify background feature trajectories, the problem we address in this work.

It is well known theorem that background trajectories tend to form a low rank matrix, a property that has been heavily explored for various applications such as stabilization and background subtraction [Sheikh et al. 2009; Cui et al. 2012; Liu et al.]. These methods assume that long-range background tracking is possible, thus are not directly applicable to videos with constant camera motion, where background features can only be tracked in short ranges.

Video stabilization For videos shot by hand-held cameras, video stabilization is an essential tool for improving their perceptual quality. To stabilize a video, estimating the camera motion is a key step. A typical video stabilization pipeline contains the following steps: (1) feature extraction and tracking/matching, for original camera motion estimation; (2) computing new, more stable camera motion; (3) applying the new camera motion to render the final frames. Previous methods largely focus on the later two steps, and study various ways to compute the new camera path, such as 2D smoothing [Litvin et al. 2003; Matsushita et al. 2006], epipolar geometry [Goldstein and Fattal 2012], 3D reconstruction [Liu et al. 2009], subspace projection [Liu et al.], L^1-optimization [Grundmann et al. 2011], and bundled camera path optimization [Liu et al. 2013b]. These methods can also be used to stabilize the display of a projector [Willi and Grundhofer 2016]. Despite using robust feature tracking [Battiato et al. 2007], these methods pay very little attention to the first step. They only use basic methods such as RANSAC or heuristic filters, or manual feature selection [Bai et al. 2014; Yang et al.] for removing feature outliers. As a result, these methods often fail on videos of highly dynamic scenes, as explicitly reported as a major limitation in the literature [Grundmann et al. 2011; Liu et al. ; Liu et al. 2013b]. As a replacement for RANSAC, our work can be easily integrated into these approaches to improve their performance in challenging cases.

3 Algorithm

3.1 Overview

As heavily explored in previous approaches, our method relies on the fact that background feature displacements at each frame can be well approximated by a global homography. Although the homography assumption is not accurate when strong parallax exists, we found it works well for background identification, a task that has a higher level of tolerance against alignment errors than other rendering applications such as hole filling. In contrast, dynamic foreground objects often exhibit far more complex motion, leading to less coherent feature trajectories. We cluster feature points based on their motion, and rely on the coherence of the feature trajectories inside each cluster to identify the background.

However, for complex videos captured with moving cameras, the background features often have very short live spans, sometimes much shorter than foreground features (see Fig. 3(a)). Motion co-

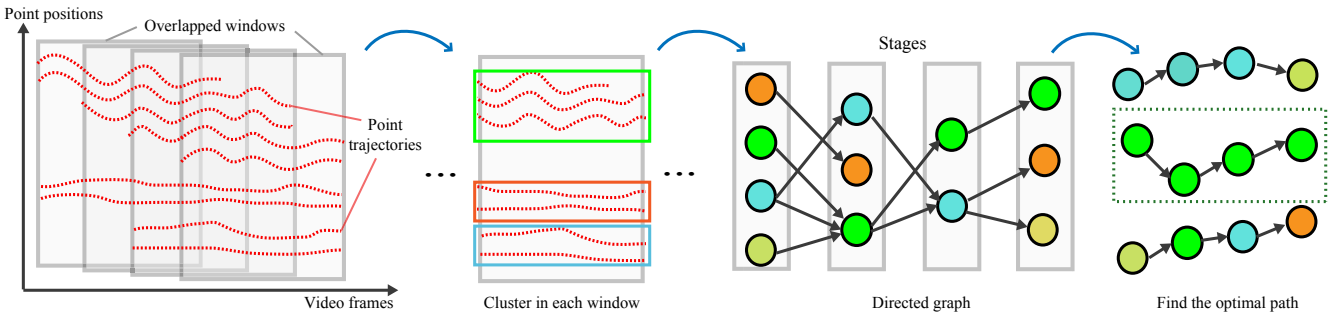

Figure 2: *Algorithm illustration.*

herence analysis thus can only applied in small local temporal windows. Applying motion analysis in short temporal windows brings additional difficulties for background identification. Firstly, in a short window, some parts of the foreground objects may undergo a similar rigid transformation to the background. Secondly, if we make independent decisions in each local window, features spanning multiple temporal windows may be assigned to conflicting labels. It is thus necessary to consider all feature trajectories together to achieve a coherent and robust background labeling.

Considering the needs for both local and global temporal feature analysis, we propose a two-step approach. As shown in Fig. 3, we first divide a video into overlapping short temporal windows. In the first step, we apply motion analysis in each local window, and group feature trajectories into a set of clusters: each cluster is a background hypothesis. In the second step, treating each feature cluster as a node and arranging them in the temporal order, we build a directed graph model, and apply global optimization to obtain a path through the graph that has the overall simplest motion. Finally, we apply a cluster refinement step to identify as much background features as possible in each local window.

3.1.1 Local motion analysis

Our method focuses on finding background features from tracked feature trajectories. Any reliable feature point detection and tracking methods can be used to initialize this process. In our implementation, we use the standard KLT tracker [Baker and Matthews 2004] given its robustness and efficiency. For the i-th tracked point p^i, its position in frame t is denoted as p^i_t. The starting and ending frame of p^i are denoted as s^i and e^i, respectively.

After feature tracking, as shown in Fig. 2, we segment the video into K overlapping temporal windows, each has W frames. The amount of overlapping is $W/2$. We use a fixed width for the windows: $W = 40$ for videos at 30fps. For features that exist on more than $0.5W$ frames in window k, we add it to the feature set P_k, which stores all the features that will participate in motion analysis in this window. We purposely avoid using feature trajectories that are too short since they are less reliable.

For each feature set P_k, we apply motion clustering, a problem well studied in the motion segmentation literature [Ma et al. 2007; Vidal et al. 2008]. For an object with rigid motion, which contains P feature trajectories with width W, following the representation in [Vidal et al. 2008], the spatial feature positions can be linked with their 3D coordinates $\{(X_p, Y_p, Z_p)\}_{p\in[1,P]}$ by:

$$\Gamma = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1P} \\ y_{11} & y_{12} & \dots & y_{1P} \\ \vdots & \vdots & \vdots & \vdots \\ x_{W1} & x_{W2} & \dots & x_{WP} \\ y_{W1} & y_{W2} & \dots & y_{WP} \end{pmatrix} = \begin{pmatrix} A_1 \\ \vdots \\ A_W \end{pmatrix} \begin{pmatrix} X_1 & \dots & X_P \\ Y_1 & \dots & Y_P \\ Z_1 & \dots & Z_P \\ 1 & \dots & 1 \end{pmatrix} \quad (1)$$

where A_t is the affine projection matrix at frame t. The rank of the left matrix should be smaller than the minimum rank of the right two matrices: ideally less than 4. If there are n moving objects (including the background), all feature trajectories can be represented by $[\Gamma_1, \Gamma_2, ..., \Gamma_n]^T$. Motion segmentation aims to decompose the trajectory matrix according to Eqn. 1. Given that there are missing entries due to occlusion, we use the method proposed by [Rao et al. 2010] to first complete the entries based on disciplined convex programming. Following [Ma et al. 2007], we then use the agglomerative lossy compression method to choose the best clustering result with the smallest recovery error from different segmentation suggestions. Some clustering examples are shown in Fig. 3.

Note that for this application, we use an affine model as an approximation to real-world cameras in camera motion analysis due to its generality and simplicity. Such approximation has been proved to be effective for motion analysis [Ma et al. 2007; Rao et al. 2010], and turns out to be also sufficient in practice for this task in small temporal windows.

3.1.2 Global optimization

For the kth local window, local motion analysis yields n_k feature clusters. To identify background ones, we build a directed graph over the entire video. Taking different windows as stages from the start to the end of the video, we define n_k nodes in each stage: each node corresponds to a feature cluster. Denote these nodes in stage k as $\{N^k_1, N^k_2, ..., N^k_{n_k}\}$. We add an arrow from one node in the current stage to another in the next stage, only if some feature trajectories exist in both clusters. The number of shared trajectories between the two nodes are denoted as $S^{u,v}_{k,k+1}$, where u and v are the labels of the nodes in stage k and $(k+1)$, respectively.

Once the directed graph is built, the goal is to find a continuous path through the entire video that is optimal according to some background measurement metrics. In the rare case where there is no single link between stage k and $k+1$ (sudden scene change or

Figure 3: *Examples of feature clustering in temporal windows. Feature points of the same color form a cluster in the current temporal window. The red arrows point to the clusters that are labelled as background using our method.*

severely blurred frames), we just divide the graph into two and run optimization in each. The key to the success of this optimization is a properly defined background metric, which we describe next.

In a dynamic video shot by a moving camera, the background trajectory matrix will have a lower rank than those of foreground feature clusters for two reasons: (1) the background motion can be approximated as a homography transformation, which is simpler than typical non-rigid, spatially-varying foreground object motion; (2) the background motion is caused by camera motion alone, while foreground motion contains both camera and object motion. We analyze the local motion complexity by checking the rank of the trajectory matrix of each graph node. For a given trajectory matrix Γ, we apply SVD decomposition:

$$\Gamma = U\Sigma V^T \tag{2}$$

If Γ is a low rank matrix, Σ will have only a few nonzero singular values. In practice we take the number of singular values which are larger than a threshold τ as the rank value. τ is set to 0.05 in our system and the feature point positions are normalized to $[0, 1]^2$. Here, we still use the low rank approximation as described in Sec. 3.1.1. For video shorter than 50 frames (2 seconds), Rao et al. [2010] and Cui et al. [2012] have shown that SVD-based rank extraction methods can well represent the potential motion complexity. If larger windows are needed (e.g., larger than 100 frames), the projective factorization method proposed in [Christy and Horaud 1996; Sturm and Triggs 1996] can be used to produce more precise rank values.

We use this rank analysis to define the weights of the edges between neighboring states. Only trajectories that appear in both neighboring nodes are used to form a matrix. The calculated approximate rank of the matrix is denoted as $r^{i,j}_{k,k+1}$, where i and j are the cluster indexes in stage k and $k+1$, respectively. The weight of the edge $e^{i,j}_{k,k+1}$ is defined as:

$$\omega^{i,j}_{k,k+1} = exp(-\alpha S^{u,v}_{k,k+1}) \cdot r^{i,j}_{k,k+1} \tag{3}$$

The smaller the weight is, the less complex is the underlying motion. Note that we add an exponential term that considers the influence of the number of shared features between two stages. When the ranks are equal between two matrices, the pair of nodes that have more features in common will have a smaller weight. In our experiments we set $\alpha = 0.1$.

Now we need to find an optimal path through which the sum of the edge weights is the smallest. The optimal path should be chosen from all possible paths from the first stage to the last one. We use dynamic programming to find the best path from a node i in the first stage to the node j in the last stage. Defining the minimum sum of the weights from node N^i_k in stage k to node N^j_l in stage l as $M(N^i_k, N^j_l)$, we have:

$$M(N^i_k, N^j_l) = min\{M(N^i_k, N^q_{l-1}) + \omega^{q,j}_{l,l-1}\}_{q\in\Phi^j_l} \tag{4}$$

Where Φ^j_l represent the set of subscripts of nodes which have arrows pointing to the node. This problem can be solved by dynamic programming. Finally, we exhaustively search different combinations of the starting and ending clusters to find the optimal path. All feature trajectories that are in the nodes of this path are labeled as background.

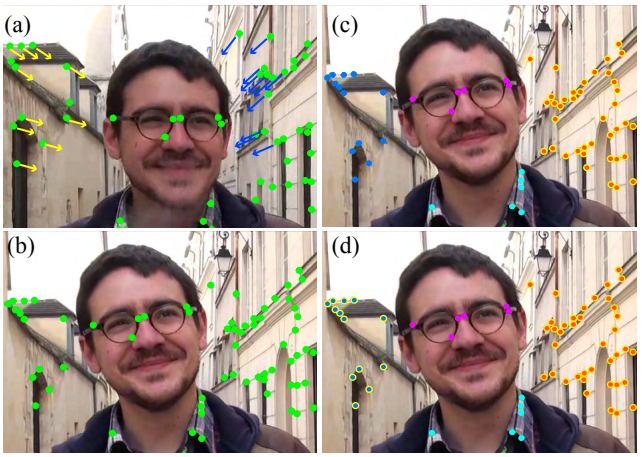

Figure 4: *(a)(b) Feature points on two frames. The arrows show the motion directions of two different groups of background points. (c) Feature clustering result. Yellow ones are selected as background in Sec. 3.1. (d) Result after label refinement in Sec. 3.2.*

In Fig. 3, we show two challenging examples for background identification. In example (a), the background is heavily occluded by moving foreground objects; the camera also sweeps, keeping the visible portion of the background changing. In example (b), the foreground object is smaller, but the water surface has a complex appearance change and a large number of feature points can be detected in it, which cannot be used for estimating the camera motion. Our method produces reliable background labeling in both cases.

Figure 5: *Examples of final selected background feature points, showing as green. Pink points are classified as non-background.*

3.2 Background label refinement

The above optimization process labels one background cluster in each temporal window. However, in some cases, the background features may be divided into several clusters in a local temporal window. As shown in Fig. 4, when the camera is zooming out, the points on the left wall move towards right, while those on the right wall move in the opposite direction. Although they are following the same homography transformation, they may still be grouped into different clusters due to their motion difference, thus only a part of the background features will be correctly labeled by the above optimization process. We propose an additional label refinement step to solve this problem, by using more features such as color and spatial position.

In each temporal window, we first exclude all features that have been labeled as background, and re-cluster the remaining features in the joint color and spatial space. For color feature, we use the average Luv color in a small neighborhood centered at a feature point in all frames that it appears. For spatial feature, we use its normalized spatial positions at the starting and ending frame of this window. We use manifold mean shift cluster [Subbarao and Meer 2006] to segment these feature points into groups. Note that these groups are different from the original clusters which are formed by using motion information only.

We then check if each feature group's motion is consistent with the existing background features. This is done by first estimating the homography matrices H_t between neighboring frames using the current background feature set B_k for the k-th window, and computing average mapping errors c_t for all frames in this temporal window. We then check the errors of using $\{H_t\}$ to estimate the positions of each point group in this window. For a feature group, if its average error is smaller than c_t, then the features in this group are added into B_k. This procedure is iteratively performed until no more point group can be added into B_k.

We choose homography as the motion model here because it is a reasonable approximation for camera motion estimation as shown in recent works on background replacement [Zhong et al. 2014] and video synchronizing [Wang et al. 2014]. For videos with smaller FOVs, the subtle motion difference among features caused by depth is usually overwhelmed by large camera translation and rotation, then a homography can well model the transformation between neighboring frames. As in Fig. 4, the feature points movements caused by camera zooming out and shaking are much larger than that caused by scene depth variation, which allows us to produce correct background labels.

Note that some long feature trajectories that live across multiple stages may be labeled only partially as background. This is a common situation when a foreground object stops moving for a short time, so that its feature points have the same motion as the background temporarily. Although this does not affect camera motion estimation much, for semantic consistency we remove these features from the final background feature point set.

4 Applications

With reliable background feature selection, our method directly leads to better camera motion estimation, a key component in many video editing tasks.

4.1 Improving video stabilization

Feature tracking and camera motion estimation is the first important step in previous video stabilization approaches. To demonstrate the effectiveness of our method, we develop an examplar video stabilizer that combines our feature selection with the widely-used L1 optimization framework [Grundmann et al. 2011; Zhang et al. 2015]. Note that our method is not limited to a specific stabilization algorithm, and can be easily combined with others. In this particular implementation, given a sequence $\{I_0, \ldots, I_n\}$, a homography transform matrix H^{t+1} is estimated between each successive pair of frames via $I^{t+1} = H^{t+1}I^t$, using selected background features extracted by our method. The original camera path is represented by the sequence of matrices $H^1, \ldots, H^n$. The objective of stabilization is to obtain an updated proxy camera path $\{H'^t\} = \{P^t H^t\}$, where the derivatives of the entries of $\{H'^t\}$ are minimum. It is obvious that in this method, accurately estimating the original homography matrices between neighboring frames is a key factor to the quality of the final stabilization result.

Comparisons and Evaluation

Fig. 6 shows a comparison between our stabilization result and a recent method proposed in [Liu et al. 2013b]. Due to the heavy influence of moving foreground people, the original camera motion cannot be reliably estimated using simple filtering used in [Liu et al. 2013b], thus their results contain jittering and wobbling. Our method generates better results with a more stable camera path. The complete sequences and more comparisons with other methods can be found in the supplementary material.

To quantitatively evaluate how much improvement our method enables, We further conduct a comprehensive evaluation on a synthetic dataset. We construct dynamic 3D scenes in Maya, and simulate both jittering camera paths to produce input videos, as well as steady camera paths to produce the ground truth for stabilization. This is done by manipulating the camera parameters including rotation, translation and zooming directly in Maya scenes. Fig. 8 show such an example.

	Ours(%)	L1-optimization(%)
Rotation-X,Y,Z	7.6, 6.3, 3.0	21.3, 30.2, 24.1
Zooming	8.9	12.5
Translation-X,Y	12.2, 13.5	18.7, 19.0
Average	8.6	21.0

Table 1: *Average errors of the camera motion parameters of the stabilized camera paths.*

In Fig. 8, we also plot the motion parameters including zooming, rotation and translation from the recovered homography sequence using the method described in [Malis and Vargas 2007]. To simulate a shaky camera, we add smaller noises in translation and zooming, and larger noises in rotation. We have found that even 1 or 2 degrees of rotation noise will cause serious shakiness. We stabilize the shaky videos using the L1-optimization stabilization method [Grundmann et al. 2011] and our modified stabilizer. We then use the ground truth background points in the stabilized result to computed the stabilized homography matrices, and compare the estimated motion parameters with the ground truth. As the curves in Fig. 8 show, compared with the path recovered without background identification, our result is more accurate, i.e., closer to the ground truth, especially when there are large moving foreground objects. Table 1 shows the average parameter recover errors on this synthetic dataset. On average the recovered parameters have an error of 8.6% after applying background identification, compared with 21% using standard RANSAC. Furthermore, our result has much smaller errors in rotation, the main source for content jittering. Visual comparisons further confirm that videos produced with background identification have higher visual quality than those without (see user study in the supplementary material).

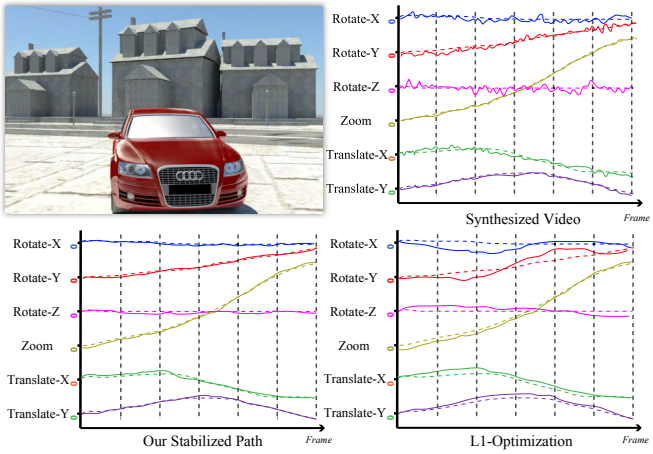

Figure 8: *One example in the synthetic stabilization dataset. The curves show the ground truth camera motion (dashed line), jittered path (top-right), our stabilized path (bottom-left) and result from L1-optimization stabilization method in [Grundmann et al. 2011]*

.

4.2 Layered video editing

Given the identified sparse background feature points using the proposed method, we can further produce pixel-wise dense background masks in dynamic videos. The dense masks are required for layered video editing, an important task in the postprocessing pipeline. To achieve this, we first use a dense video segmentation method proposed in [Grundmann et al. 2010] to over-segment the input video. For regions containing labeled feature points, they are directly labeled as either foreground or background, unless there are conflicts among feature labels inside the region. The labels are then propagated using a greedy method to other spatial and temporal neighboring regions that do not contain tracked feature points. The similarities of the average optical flow and pixel color are used to determine which label to assign, if an unlabeled region is adjacent to both foreground and background regions. The labels will be iteratively propagated to the unlabeled regions if the labeled region has more similar temporal and spatial features.

The dense masks can be used in many editing tasks, for example, inserting a new object in the background. The user can place the new object at the proper location on a key frame. Using the labeled background feature points, we can compute a local homography transformation between a pair of successive frames, allowing the new object to change its position across frames according to the camera motion. For seamless blending when the foreground occludes the new object, we use a video matting method [Bai et al. 2011] to produce soft masks for the foreground layer in such frames for compositing. An example is shown in Fig. 7, where correct foreground occlusion is generated.

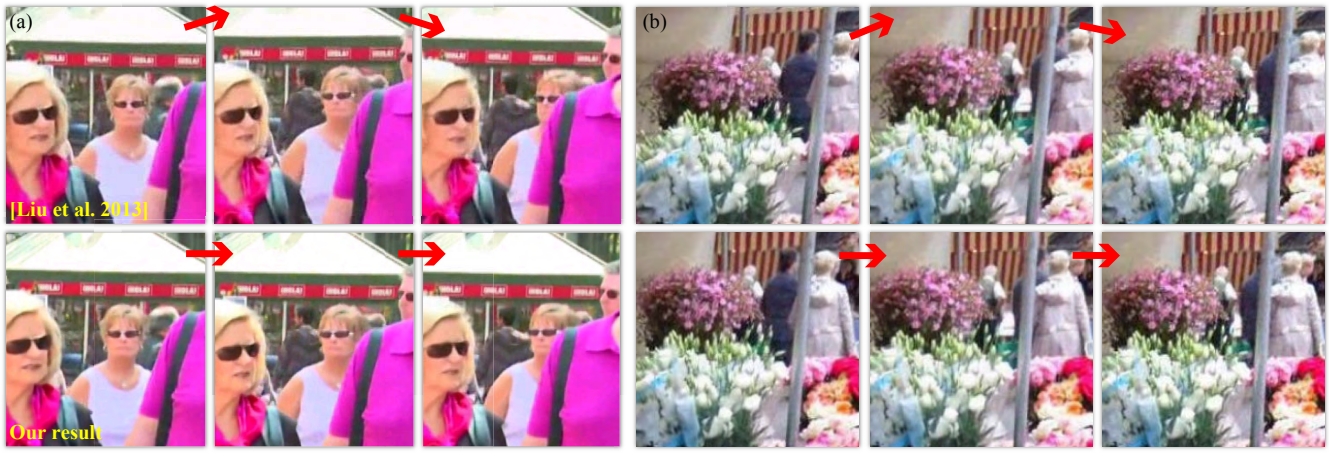

Figure 6: *Comparisons on video stabilization. Top: results of Liu et al. [2013b]. their result contains jittering and unnatural deformation. Bottom: our results do not contain these artifacts. Please refer to the supplemental material for full video results.*

Figure 7: *Examples of background edit propagation. The arrows show the edited regions. Please refer to the supplemental material for full video results.*

4.3 Background reconstruction

Constructing a clean background plate from a video is another common video editing task. For a dynamic video with moving foreground objects, different parts of the background will appear at different times. Using the labeled background features produced by our method, we can (1) better align video frames into a global coordinate space; and (2) determine which region to take on each frame in order to produce the final background plate. Again, we use the sparse-to-dense propagation method introduced in Sec. 4.2 to produce the dense mask for all the background region in different frames. To further improve the sharpness of the reconstructed background, we introduce an extra refinement step by performing standard guided patch synthesis, i.e., querying background image patches from original video frames to reconstruct the initial background. Every frame can then be aligned to the global background plate to generate a background video, as shown in Fig. 9 and the supplementary video. In comparison, replacing our method with RANSAC results in a background that contains obvious distortions and foreground residuals, as shown in Fig. 9.

5 Results and Evaluation

5.1 Performance

We implemented our approach in a mix of C++ and Matlab on a PC with an Intel Xeon E5620 CPU at 2.4GHz and 16GB RAM. Using a single core, for a video with 240 frames, feature tracking takes about 10 seconds, motion segmentation takes about 1 to 8 minutes in each temporal window according to the number of feature points involved, using the Matlab implementation provided by Rao et al. [2010]. It takes $3-5$ seconds to compute the rank of each cluster and find the optimal path. The final refinement step takes about 10 seconds to converge in each window. Note that all these steps are fully automatic. We have also parallelized motion segmentation and final refinement so multiple temporal windows can be computed simultaneously. The whole algorithm has a linear complexity over the video length.

5.2 Comparisons

RANSAC is a common method used in previous camera motion estimation approaches for outlier removal [Baker and Matthews 2004]. We use a 5-point-model RANSAC to filter the outliers and

Figure 9: *Using our method for background reconstruction. The global background plate recovered using our method is shown on the top. The background plate recovered using the inliers of RANSAC is shown on the bottom, which contains obvious distortion and foreground residual.*

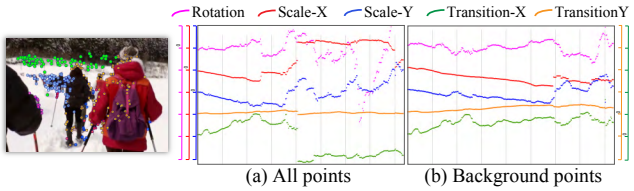

Figure 11: *Comparing homography parameters estimated with and without our background feature selection. (a) RANSAC filtering on all feature points. (b) RANSAC filtering on selected background features. The curves show the parameters of homographies computed between each neighboring frame pair.*

directly estimate a full homography transformation to find correspondence between frames. As discussed earlier, it does not work well when the background features are actually minority in the whole feature set, such as the example shown in Fig. 11. It visualizes the transformation parameters of the estimated homographies using RANSAC, with and without our background identification. It shows that using RANSAC only, the transformations are unreliable and inconsistent between neighboring frames (Fig. 11(a)). Applying the proposed method prior to RANSAC allows us to estimate the correct camera motion transformations.

In the recent camera path improvement approach [Zhang et al. 2015], a video saliency map is used to exclude possible foreground features with high saliency values. The saliency is computed from both appearance and motion contrasts. However, as shown in Fig. 10, when the moving objects are large, some parts of the moving objects may have small saliency values (e.g., the wheels on the bike), which will be mistakenly labeled as background by this approach.

Motion segmentation methods [Rao et al. 2010; Tuzel et al. 2005] can cluster feature trajectories into groups. Using motion segmentation, one can first segment feature trajectories, and then choose the group that has the lowest rank to be the background. To compare with this strategy, we first use the method in [Rao et al. 2010] to complete the missing entries in the trajectory matrix using CVX, and perform motion segmentation on the full trajectories of all feature points. The result is shown in Fig. 10(c) (full videos are in the supplementary materials). For such challenging cases, feature completion is not reliable because most of the trajectories are short, leading to erroneous motion segmentation results. In contrast, our method can effectively handle both short and long feature trajectories, resulting in more accurate and stable background identification, as shown in Fig. 10 (d).

5.3 Quantitative evaluation I

We conduct a quantitative evaluation on the proposed method over highly dynamic videos. We collect 10 home videos with large camera motions, and extract features from them. We develop a user interface to allow human labelers to manually label background features using paint brushes. The labeling is first done on keyframes, then automatically propagated by feature matching. The labelers are required to carefully examine the result on each single frame and correct errors if spotted. We recruited 5 labelers to label each video. For each feature trajectory, it is labeled as background only if the majority of labelers agree. This forms our benchmark dataset. It contains about 52,000 feature point trajectories, in which 8,100 are labelled as background.

We compare our method with RANSAC, the saliency-based method [Zhang et al. 2015] and the motion segmentation method [Rao et al. 2010]. Brox and Malik [2010a] proposed another robust motion segmentation method, but it does not show much advantage over the ALC method proposed by Rao et al. [2010] for short videos. We only compare with the ALC method. For RANSAC, we perform it to estimate a homography between neighboring frames, and treat the selected inliers as background. The performances of different methods are shown in Tab. 2. The results suggest that our method achieves significantly more accurate results than all alternatives. The accuracy and recall of RANSAC are both low, as it tends to select foreground features when moving objects are large. We also try a hybrid approach to only apply RANSAC on the background features selected by our method, which yields higher accuracy but lower recall, compared with using our method only. It suggests that for applications where accuracy is more critical, one can choose to apply RANSAC as an optional post-processing step to our method.

Method	Accuracy(%)	Recall(%)	F-score
Our method	92.1	90.1	0.911
Ours + RANSAC	97.7	83.8	0.902
RANSAC	56.4	49.5	0.533
Saliency	64.6	69.3	0.670
Rao 2010	46.8	79.0	0.588

Table 2: *Recall and precision of different methods in Quantitative evaluation I.*

5.4 Quantitative evaluation II

We conduct another quantitative comparison between our methods and existing point trajectories segmentation methods, We use dynamic videos from the VSB-100 video segmentation benchmark [Galasso et al. 2013] and DAVIS dataset proposed in [Perazzi et al. 2016] as our evaluation dataset. In VSB-100, for each HD

(a) Video saliency map (b) Result from saliency (c) Result from [Rao et al. 2010] (d) Our result

Figure 10: *Comparisons on background feature selection. Background features are shown in green and foreground ones are in purple. (a) Saliency maps for two different frames, computed using [Zhang et al. 2015]. (b) Selected background features by thresholding the saliency map. (c) Result of motion segmentation [Rao et al. 2010]. (d) Our result.*

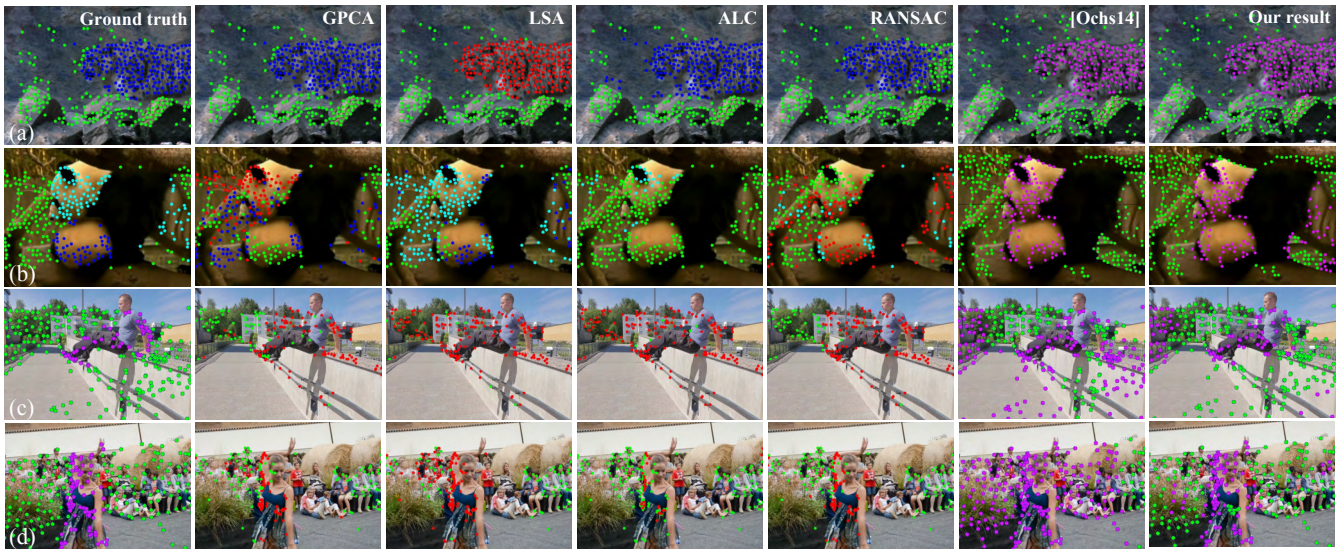

Figure 12: *Comparisons on the VSB-100 dataset (a, b) and DAVIS dataset (c, d). Background points are shown in green.*

quality video, there are pixel-wise manual labels for each object, provided on one keyframe in every 20 frames. However there is no foreground or background labels. To produce ground truth background labeling, we further provide manual annotations of background objects on these keyframes. Given that not all videos in the VSB dataset are suitable for evaluating our task (e.g., videos shot by static cameras), we choose 31 videos that contain significant camera and object motion, yielding about $96,000$ feature trajectories overall. For videos from DAVIS, we directly use the ground truth for background segmentation to label all the feature trajectories, generating $130,000$ feature trajectories.

We compare our method against representative motion segmentation methods, including GPCA [Vidal et al. 2008], LSA [Yan and Pollefeys 2006], RANSAC [Tron and Vidal 2007], and the method proposed by Rao et al. [Rao et al. 2010]. Except for the last one, these methods cannot handle incomplete point trajectories. We thus complete the missing entries in the trajectory matrix before applying these methods. Furthermore, these methods only produce feature clustering results. We thus apply an "oracle" classifier to classify each cluster: supposing that background features take a proportion of τ in all extracted features, if more than τ of the features in a cluster are marked as background in the ground truth, we label the cluster as background. We also compare our method with the latest dense moving object segmentation method [Ochs et al. 2014]. where the segmentation masks are used to identify foreground feature points.

The accuracy and recall of different methods are shown in the Tab. 3. The results suggest that previous motion segmentation methods often group nearby foreground and background features into same clusters (see Fig. 12), while our method generates a much better separation between them. Note that in this dataset, many objects with very little motion (like standing people) are labelled as foreground in the ground truth. Since these objects present little object motion, they are mostly identified as background by our method, which lowers our performance scores compared with experiment I. The results from [Ochs et al. 2014] are good when the camera moves slowly in a small range. However, when the camera moves fast as in the examples shown in Fig. 12(c), it will miss a large portion of background points. That is because their method focuses on where optical flow estimation works best, so background

Figure 13: *A failure case where the camera is mounted on the moving object, thus the moving object is classified as background.*

regions that are updating fast will be identified as foreground due to erroneous flow estimation.

Method	Accuracy(%)		Recall(%)		F-score	
	VSB	DAVIS	VSB	DAVIS	VSB	DAVIS
Ours	87.1	84.2	87.6	82.4	0.874	0.833
GPCA	67.5	71.0	70.1	69.0	0.688	0.700
RANSAC	66.2	68.3	70.8	66.0	0.684	0.672
LSA	66.8	72.2	67.5	68.2	0.671	0.702
Rao2010	69.6	67.3	78.9	62.8	0.740	0.650
Ochs2014	66.9	80.9	70.9	80.7	0.689	0.808

Table 3: *Recall and precision of different methods in Quantitative evaluation II.*

5.5 Limitations and discussions

Our method may fail in some special situations. Firstly, if the camera moves together with the foreground object (often called a "Tracking Shot"), the foreground object will appear to be relatively still in the video, and its feature trajectories will form a low rank matrix, which are likely to be labeled as background using our method. Fig. 14 and supplementary video show such a failure case. In this example, since the camera is mounted on the foreground, the moving object is classified as the background.

Secondly, if the foreground object occupies the entire frame, or the camera is moving too fast at one moment and no background feature can be tracked, the computed background path will be divided into disconnected chunks. The global optimization would break because there is no path that can last from the beginning to the end. Some extra components need to be added to this algorithm in order to handle more special cases. For example, when the background completely disappears, we can add long-term feature registering. For stationary foreground, semantic recognition can be added to help identify the background.

6 Conclusion

We have presented a new background identification method to extract background feature points from videos of highly dynamic scenes, a fundamental building block in many computer graphics and vision tasks. The results can be used to significantly improve camera path estimation, and can be used in many video editing and enhancement applications, such as video stabilization, background reconstruction and video object composition. As future work, we plan to further improve the method by introducing semantic parsing on different regions, and explore other applications such as 3D reconstruction from dynamic scene. We also plan to explore learning-based methods for background identification, by establishing a large dataset of dynamic videos and applying state-of-the-art machine learning techniques on it.

7 Acknowledgments

This work was supported by the Natural Science Foundation of China (Project No. 61521002, 61133008), the General Financial Grant from the China Postdoctoral Science Foundation (Grant No. 2015M580100), Research Grant of Beijing Higher Institution Engineering Research Center, and Tsinghua University Initiative Scientific Research Program.

References

AREV, I., PARK, H. S., SHEIKH, Y., HODGINS, J., AND SHAMIR, A. 2014. Automatic editing of footage from multiple social cameras. *ACM Trans. Graph. (SIGGRAPH 2014) 33*, 4, 81.

BAI, X., WANG, J., AND SIMONS, D. 2011. Towards temporally-coherent video matting. In *Mirage*.

BAI, J., AGARWALA, A., AGRAWALA, M., AND RAMAMOORTHI, R. 2014. User-assisted video stabilization. *Computer Graphics Forum (EGSR 2014) 33*, 4, 61–70.

BAKER, S., AND MATTHEWS, I. 2004. Lucas-Kanade 20 years on: A unifying framework. *IJCV 56*, 3, 221–255.

BARNICH, O., AND VAN DROOGENBROECK, M. 2009. Vibe: A powerful random technique to estimate the background in video sequences. In *IEEE ICASSP*, 945–948.

BATTIATO, S., GALLO, G., PUGLISI, G., AND SCELLATO, S. 2007. Sift features tracking for video stabilization. In *Int. Conf. Image Analysis and Processing*, 825–830.

BOULT, T. E., AND BROWN, L. G. 1991. Factorization-based segmentation of motions. In *IEEE Workshop on Visual Motion*, IEEE, 179–186.

BROX, T., AND MALIK, J. 2010. Object segmentation by long term analysis of point trajectories. In *ECCV*. Springer, 282–295.

BROX, T., AND MALIK, J. 2010. Object segmentation by long term analysis of point trajectories. In *ECCV*. Springer, 282–295.

CHANG, J., WEI, D., AND FISHER III, J. W. 2013. A video representation using temporal superpixels. In *IEEE CVPR*, 2051–2058.

CHEN, B.-Y., LEE, K.-Y., HUANG, W.-T., AND LIN, J.-S. 2008. Capturing intention-based full-frame video stabilization. *Computer Graphics Forum 27*, 7, 1805–1814.

CHEN, T., ZHU, J.-Y., SHAMIR, A., AND HU, S.-M. 2013. Motion-aware gradient domain video composition. *IEEE Transactions on Image Processing 22*, 7, 2532–2544.

CHENG, L., GONG, M., SCHUURMANS, D., AND CAELLI, T. 2011. Real-time discriminative background subtraction. *IEEE Transactions on Image Processing 20*, 5, 1401–1414.

CHIEN, S.-Y., MA, S.-Y., AND CHEN, L.-G. 2002. Efficient moving object segmentation algorithm using background registration technique. *IEEE TCSVT 12*, 7 (Jul), 577–586.

CHIU, C.-C., KU, M.-Y., AND LIANG, L.-W. 2010. A robust object segmentation system using a probability-based background extraction algorithm. *IEEE TCSVT 20*, 4 (April), 518–528.

CHO, S., WANG, J., AND LEE, S. 2012. Video deblurring for hand-held cameras using patch-based synthesis. *ACM Trans. Graph. (SIGGRAPH 2012) 31*, 4, 64.

Figure 14: *More background identification results. Background features are shown in green and others are in purple.*

CHRISTY, S., AND HORAUD, R. 1996. Euclidean shape and motion from multiple perspective views by affine iterations. *IEEE TPAMI 18*, 11 (Nov.), 1098–1104.

COSTEIRA, J. P., AND KANADE, T. 1998. A multibody factorization method for independently moving objects. *IJCV 29*, 3, 159–179.

CUI, X., HUANG, J., ZHANG, S., AND METAXAS, D. N. 2012. Background subtraction using low rank and group sparsity constraints. In *ECCV*. Springer, 612–625.

ELGAMMAL, A., HARWOOD, D., AND DAVIS, L. 2000. Non-parametric model for background subtraction. In *ECCV*, Springer, 751–767.

FRAGKIADAKI, K., AND SHI, J. 2011. Detection free tracking: Exploiting motion and topology for segmenting and tracking under entanglement. In *IEEE CVPR*, 2073–2080.

GALASSO, F., NAGARAJA, N., CARDENAS, T., BROX, T., AND SCHIELE, B. 2013. A unified video segmentation benchmark: Annotation, metrics and analysis. In *IEEE ICCV*, 3527–3534.

GLEICHER, M. L., AND LIU, F. 2008. Re-cinematography: Improving the camerawork of casual video. *ACM TOMCCA 5*, 1, 2.

GOLDSTEIN, A., AND FATTAL, R. 2012. Video stabilization using epipolar geometry. *ACM Trans. Graph. (SIGGRAPH 2012) 31*, 5, 126:1–10.

GRUNDMANN, M., KWATRA, V., HAN, M., AND ESSA, I. 2010. Efficient hierarchical graph based video segmentation. *IEEE CVPR*.

GRUNDMANN, M., KWATRA, V., AND ESSA, I. 2011. Auto-directed video stabilization with robust l1 optimal camera paths. In *IEEE CVPR*, 225–232.

HAYMAN, E., AND EKLUNDH, J.-O. 2003. Statistical background subtraction for a mobile observer. In *IEEE ICCV*, vol. 1, 67–74.

JIA, Y.-T., HU, S.-M., AND MARTIN, R. R. 2005. Video completion using tracking and fragment merging. *The Visual Computer 21*, 8–10.

LITVIN, A., KONRAD, J., AND KARL, W. C. 2003. Probabilistic video stabilization using kalman filtering and mosaicing. In *Electronic Imaging*, 663–674.

LIU, F., GLEICHER, M., WANG, J., JIN, H., AND AGARWALA, A. Subspace video stabilization. *ACM Trans. Graph. (SIGGRAPH 2011) 30*, 1, 15:1–10.

LIU, F., GLEICHER, M., JIN, H., AND AGARWALA, A. 2009. Content-preserving warps for 3D video stabilization. *ACM Trans. Graph. (SIGGRAPH Asia 2009) 28*, 3, 44.

LIU, F., NIU, Y., AND JIN, H. 2013. Joint subspace stabilization for stereoscopic video. In *IEEE ICCV*, 73–80.

LIU, S., YUAN, L., TAN, P., AND SUN, J. 2013. Bundled camera paths for video stabilization. *ACM Trans. Graph. (SIGGRAPH 2013) 32*, 4, 78.

LUO, D., AND HUANG, H. 2014. Video motion segmentation using new adaptive manifold denoising model. In *IEEE CVPR*, 65–72.

MA, Y., DERKSEN, H., HONG, W., AND WRIGHT, J. 2007. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE TPAMI 29*, 9, 1546–1562.

MALIS, E., AND VARGAS, M. 2007. Deeper understanding of the homography decomposition for vision-based control.

MATSUSHITA, Y., OFEK, E., GE, W., TANG, X., AND SHUM, H.-Y. 2006. Full-frame video stabilization with motion inpainting. *IEEE TPAMI 28*, 7, 1150–1163.

MUMTAZ, A., ZHANG, W., AND CHAN, A. B. 2014. Joint motion segmentation and background estimation in dynamic scenes. In *IEEE CVPR*, 368–375.

OCHS, P., MALIK, J., AND BROX, T. 2014. Segmentation of moving objects by long term video analysis. *IEEE TPAMI 36*, 6, 1187–1200.

PAPAZOGLOU, A., AND FERRARI, V. 2013. Fast object segmentation in unconstrained video. In *IEEE ICCV*, 1777–1784.

PERAZZI, F., WANG, O., GROSS, M., AND SORKINE-HORNUNG, A. 2015. Fully connected object proposals for video segmentation. In *IEEE ICCV*, 3227–3234.

PERAZZI, F., PONT-TUSET, J., MCWILLIAMS, B., GOOL, L. V., GROSS, M., AND SORKINE-HORNUNG, A. 2016. A benchmark dataset and evaluation methodology for video object segmentation. In *IEEE CVPR*, 724–732.

RAO, S., TRON, R., VIDAL, R., AND MA, Y. 2010. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE TPAMI 32*, 10, 1832–1845.

ROSTEN, E., PORTER, R., AND DRUMMOND, T. 2010. Faster and better: A machine learning approach to corner detection. *IEEE TPAMI 32*, 1, 105–119.

SAND, P., AND TELLER, S. 2008. Particle video: Long-range motion estimation using point trajectories. *IJCV 80*, 1, 72–91.

SHEIKH, Y., JAVED, O., AND KANADE, T. 2009. Background subtraction for freely moving cameras. In *IEEE ICCV*, 1219–1225.

SHI, J., AND TOMASI, C. 1994. Good features to track. In *IEEE CVPR*, 593–600.

STURM, P., AND TRIGGS, B. 1996. A factorization based algorithm for multi-image projective structure and motion. In *ECCV*, 709–720.

SUBBARAO, R., AND MEER, P. 2006. Nonlinear mean shift for clustering over analytic manifolds. In *IEEE CVPR*, vol. 1, 1168–1175.

SUN, D., ROTH, S., AND BLACK, M. J. 2010. Secrets of optical flow estimation and their principles. In *IEEE CVPR*, 2432–2439.

TAYLOR, B., KARASEV, V., AND SOATTO, S. 2015. Causal video object segmentation from persistence of occlusions. In *IEEE CVPR*, 4268–4276.

TOMASI, C., AND KANADE, T. 1992. Shape and motion from image streams under orthography: a factorization method. *IJCV 9*, 2, 137–154.

TRON, R., AND VIDAL, R. 2007. A benchmark for the comparison of 3-d motion segmentation algorithms. In *IEEE CVPR*, IEEE, 1–8.

TUZEL, O., SUBBARAO, R., AND MEER, P. 2005. Simultaneous multiple 3d motion estimation via mode finding on lie groups. In *IEEE ICCV*, vol. 1, 18–25.

VAN DEN BERGH, M., BOIX, X., ROIG, G., DE CAPITANI, B., AND VAN GOOL, L. 2012. Seeds: Superpixels extracted via energy-driven sampling. In *ECCV*, Springer, 13–26.

VIDAL, R., TRON, R., AND HARTLEY, R. 2008. Multiframe motion segmentation with missing data using powerfactorization and gpca. *IJCV 79*, 1, 85–105.

WANG, O., SCHROERS, C., ZIMMER, H., GROSS, M., AND SORKINE-HORNUNG, A. 2014. Videosnapping: Interactive synchronization of multiple videos. *ACM Trans. Graph. (SIGGRAPH 2014) 33*, 4 (July), 77:1–77:10.

WEXLER, Y., SHECHTMAN, E., AND IRANI, M. 2004. Space-time video completion. In *IEEE CVPR*, vol. 1, I.120.

WEXLER, Y., SHECHTMAN, E., AND IRANI, M. 2007. Space-time completion of video. *IEEE TPAMI 29*, 3, 463–476.

WILLI, S., AND GRUNDHOFER, A. 2016. Spatio-temporal point path analysis and optimization of a galvanoscopic scanning laser projector. *IEEE Transactions on Visualization and Computer Graphics PP*, 99, 1–8.

WU, Y., ZHANG, Z., HUANG, T. S., AND LIN, J. Y. 2001. Multi-body grouping via orthogonal subspace decomposition. In *IEEE CVPR*, vol. 2, IEEE, II–252.

YAN, J., AND POLLEFEYS, M. 2005. A factorization-based approach to articulated motion recovery. In *IEEE CVPR*, vol. 2, IEEE, 815–821.

YAN, J., AND POLLEFEYS, M. 2006. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV*. Springer, 94–106.

YANG, M., PEI, M., WU, Y., AND JIA, Y. Learning online structural appearance model for robust object tracking. *Sci China Inf Sci 58*, 3, 1–14.

ZHANG, Y., TANG, Y.-L., AND CHENG, K.-L. Efficient video cutout by paint selection. *Journal of Computer Science and Technology 30*, 3, 467–477.

ZHANG, G., JIA, J., XIONG, W., WONG, T.-T., HENG, P.-A., AND BAO, H. 2007. Moving object extraction with a hand-held camera. In *IEEE ICCV*, 1–8.

ZHANG, F.-L., WANG, J., ZHAO, H., MARTIN, R. R., AND HU, S.-M. 2015. Simultaneous camera path optimization and distraction removal for improving amateur video. *IEEE Transactions on Image Processing 25*, 12, 5982–5994.

ZHONG, F., YANG, S., QIN, X., LISCHINSKI, D., COHEN-OR, D., AND CHEN, B. 2014. Slippage-free background replacement for hand-held video. *ACM Trans. Graph. (SIGGRAPH Asia 2014) 33*, 6, 30:1–11.

ZHOU, H., YUAN, Y., AND SHI, C. 2009. Object tracking using sift features and mean shift. *Computer vision and image understanding 113*, 3, 345–352.