Spatio-Temporal Alignment of Non-Overlapping Sequences from Independently Panning Cameras

S. Morteza Safdarnejad
Michigan State University
East Lansing, MI, USA
safdarne@egr.msu.edu

Xiaoming Liu
Michigan State University
East Lansing, MI, USA
liuxm@cse.msu.edu

Abstract

This paper addresses the problem of spatio-temporal alignment of multiple video sequences. We identify and tackle a novel scenario of this problem referred to as Non-overlapping Sequences (NOS). NOS are captured by multiple freely panning handheld cameras whose field of views (FOV) might have no direct spatial overlap. With the popularity of mobile sensors, NOS rise when multiple cooperative users capture a public event to create a panoramic video, or when consolidating multiple footages of an incident into a single video. To tackle this novel scenario, we first spatially align the sequences by reconstructing the background of each sequence and registering these backgrounds, even if the backgrounds are not overlapping. Given the spatial alignment, we temporally synchronize the sequences, such that the trajectories of moving objects (e.g., cars or pedestrians) are consistent across sequences. Experimental results demonstrate the performance of our algorithm in this novel and challenging scenario, quantitatively and qualitatively.

1. Introduction

Spatio-temporal alignment of multiple videos [7–9, 11, 14, 21, 24, 28, 31] is a well-studied vision problem with a wide range of applications, e.g., human action recognition [25, 29], video editing [31], markerless motion capture [14], video mosaicing, change detection [8], and abandoned object detection [16]. Previous works study different aspects and scenarios of the spatio-temporal alignment. Some works target sequences from the *same* scene but *different* viewpoints [14, 21]. Some can handle sequences recorded at *different* times by independent moving cameras that follow a *similar* trajectory [9,11,31]. The seminal work of Caspi and Irani [7] studies spatially non-overlapping sequences when two fixed cameras move *jointly* in space.

Our work covers a novel unexplored aspect of spatio-temporal alignment of sequences, for *non-overlapping sequences* (NOS). Targeted NOS are captured by *freely* and *independently* panning cameras, from *nearby* viewpoints, with limited translation, especially in the optical axis direction. In NOS, sequences might not have any pair of frames that have spatial overlap and belong to the same world time instant. More interestingly, sequences might even not cover some common regions of the same scene over the progression of time. In other words, if we reconstruct the observed background by these sequences, the backgrounds may be non-overlapping, i.e., in Fig. 1 (a), the overall spatial overlap does not exist.

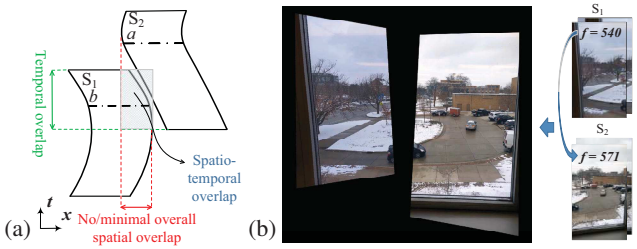

Figure 1. (a) Top view of spatio-temporal FOV of two moving cameras capturing sequences S_1 and S_2; Non-overlapping sequences (NOS) may not even cover a common spatial region over the progression of time, i.e, no overall spatial overlap exists. (b) Spatio-temporal alignment of NOS results in displaying sequences in a common coordinate and at the correct time shift.

Given the ubiquitousness of smartphones and wearcams, NOS are increasingly common. When amateur users unsynchronizedly shoot videos of an event, aligning these videos leads to a single comprehensive video, with greater spatial and temporal spans (Fig. 1 (b)). This resultant video is essentially a panoramic video, shot by smartphones, without the need to fix the cameras to each other or use tripods. Further, when many witnesses capture videos during crime actions or violations, each sequence may cover part of the story. Aligning these videos into a unified *large-scale* 3D volume provides a better grasp of the full picture.

The existing spatio-temporal alignment algorithms fail in the case of NOS, since even if there is some overall

spatial overlap, spatial alignment of apparently overlapping frames, e.g., frames a and b in Fig. 1 (a), obviously violates the temporal alignment. However, by decomposing the task to spatial alignment first and then temporal alignment based on scene dynamics, the problem can be solved. In general our proposed algorithm assumes NOS satisfy the following two assumptions. 1) Although the sequences have no corresponding frames that share a common scene at the same world time stamp, and no overall overlap as in Fig. 1 (a), they cover nearby parts of a scene from similar view angles. 2) There are moving objects in the scene which move from the field of view (FOV) of one camera to FOV of other cameras. Note that in panoramic imaging the best results are obtained if the camera has nodal camera motion, otherwise parallax-tolerant methods should be used to hide artifacts from parallax [15, 32]. Similarly, our proposed algorithm creates the best result if the camera baseline is small, although due to the non-overlapping videos, larger baselines are handled with minor visual degradation.

Our algorithm utilizes global motion compensation to map each frame to a camera-motion-removed video and reconstruct the background for each sequence, independently. With the two assumptions, these potentially *non-overlapping* backgrounds are aligned via appearance cues and also the prediction that ***where*** a moving object leaving FOV of a camera will appear in FOV of another camera. Collection of the former mappings and the latter background alignment, can spatially align each frame of each sequence with respect to frames from other sequences. Given the spatial alignment and the assumption 2, we predict ***when*** a moving object leaving FOV of one camera will appear in FOV of another. We mathematically formulate this prediction and estimate the temporal alignment.

In summary, this paper makes these contributions:

◇ A new scenario in spatio-temporal alignment of sequences is identified and studied.

◇ A spatial alignment algorithm for NOS via alignment of non-overlapping reconstructed backgrounds and consistency of objects movement is proposed.

◇ The trajectory of moving objects with smooth path is used as a clue for temporal alignment of NOS.

2. Previous Work

The prior works in spatio-temporal alignment of sequences mostly differ in their *assumptions* and *scenarios*, e.g., the camera movement (static, jointly moving, or moving), camera view-point (similar or distinct), extent of overlap in sequences, and extent of similarity of camera motion paths. The work of [12] presents an excellent taxonomy of these assumptions, one of which is that, to align sequences from the same event captured by freely moving cameras, coherent scene appearance is assumed. We lift this assumption by handling non-overlapping sequences, although we do assume negligible camera movement in the optical axis direction. We now review key scenarios in prior work.

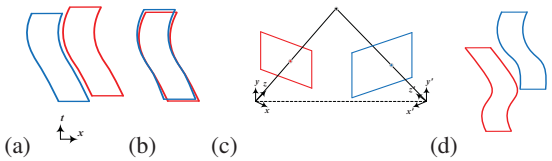

Figure 2. Various scenarios in spatio-temporal alignment of sequences: (a) jointly moving cameras, (b) independently moving cameras at different times following similar trajectories, (c) stationary cameras with different viewpoints, (d) the proposed independently panning cameras with non-overlapping sequences.

Jointly moving cameras Caspi and Irani align spatially non-overlapping sequences when two closely *attached* cameras move *jointly* in space (Fig. 2 (a)) [7]. Assuming cameras share the *same* projection center, their relationship is modeled as a fixed homography $\mathbf{H}$. Esquivel et al. [10] relax the projection center assumption and calibrate a multi-camera rig from non-overlapping views, assuming synchronized sequences. Recently, some works generate panoramic videos from synchronized sequences of a multi-camera rig [15,22]. Given that the cameras are fixed relative to each other and have overlapping FOV, these works concentrate on removing the parallax artifacts. In contrast, we focus on creating video panoramas of *unsynchronized* sequences from *independently panning* cameras, *removing* the requirement of joint cameras and overlapping sequences.

Cameras following similar trajectories The authors of [9, 11, 12, 31] align sequences recorded at *different* times by independent moving cameras that follow a *similar* trajectory (Fig. 2 (b)). Assuming one sequence is entirely contained (temporally) within the other, in [9], the alignment is formulated as an energy minimization alternately solved for temporal and spatial alignment parameters and is evaluated on four sets of real videos. In [31] an interactive method for nonlinear temporal video alignments is proposed for video editing. All these methods require coherent scene appearance and cannot handle sequences from moving cameras with no overlap in FOV — the targeted scenario of NOS.

Stationary cameras at different views Padua et al. [21] target sequences from the *same* scene but *different* viewpoints (Fig. 2 (c)). The *stationary* cameras allow the estimated camera's epipolar geometry remain fixed. Motion trajectories are used as cues for both spatial and temporal alignment. Experimental results are provided for five sequences. For each sequence, the optimal tracker is chosen based on the application in hand.

Time synchronization Assuming the known 3D object location and calibrated stationary cameras, [6] synchronizes non-overlapping sequences of these cameras. Gaspar et al. [13] synchronize sequences from independently moving cameras, assuming known intrinsic parameters and visibility of two rigid moving objects in both sequences.

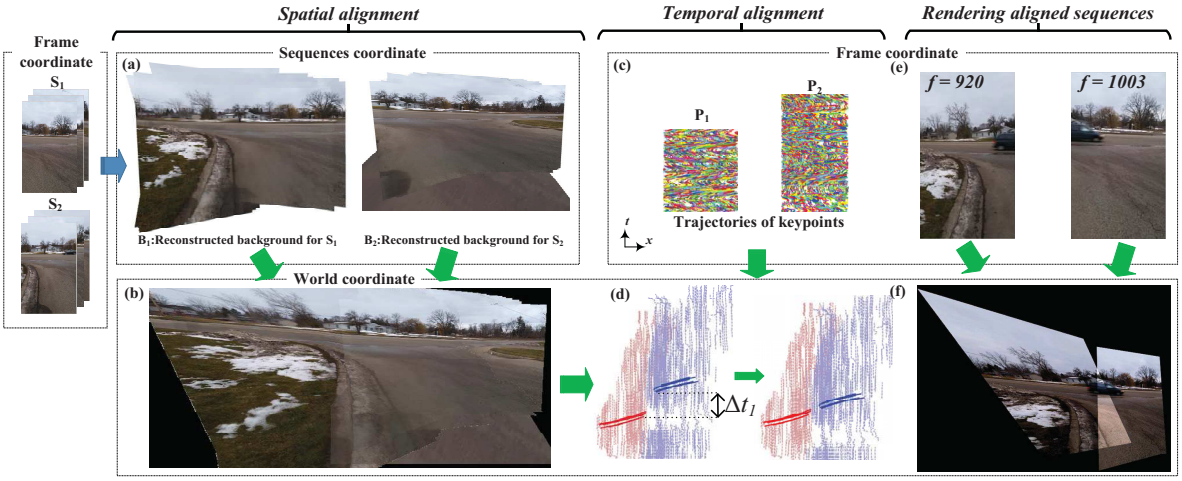

Figure 3. Flowchart of our spatio-temporal alignment algorithm. First, spatial alignment is performed by background reconstruction for each sequence (a) and aligning the backgrounds (b). Second, given the spatial alignment parameters, keypoint trajectories (c) are mapped to the world coordinate and the best temporal alignment in terms of continuity of moving object trajectories is found (d). Finally, spatio-temporal alignment parameters are used for displaying the sequence in a world coordinate system and at the correct time shift (e).

Lu and Mandal [19] model the video temporal alignment as a spatio-temporal discrete trajectory alignment problem. Moving objects provide the main cue in photo sequencing as well [4], whereas static objects help to find the relationship between the photos. Our method also relies on existence of at least one moving object for temporal synchronization. In fact, without spatial overlap between FOVs, any temporal alignment algorithm has to track moving objects or egomotion [12]. Our strength is that we can work with NOS where the same moving object is not visible at the same time in all sequences, without relying on camera calibration or known moving object location.

3. Proposed Method

We discussed the assumptions for the proposed spatio-temporal alignment of NOS in Sec. 1. The intrinsic and extrinsic camera parameters are not required. Also, the cameras might start capturing at different times, i.e. unsynchronizedly, with possibly distinct frame rates, and are panned freely and independently. However, best results are achieved by a small camera baseline and limited translation of cameras, especially in the optical axis direction, as we rely on global motion compensation and large variations in scale, degrade the resultant video in its rendering phase.

The proposed algorithm has two stages, (1) spatial alignment (Fig. 3(b)), which relies on the reconstructed backgrounds' appearance and consistency of movement of objects across the sequences, (2) temporal alignment (Fig. 3(d)), which uses the continuity of objects' trajectories to synchronize the videos.

Notations As shown in Fig. 3, *frame coordinate* refers to the pixel coordinate in the input video, *sequence coordinate* to the global coordinate of the reconstructed background of one video, and *world coordinate* to the global coordinate of *all* input videos where the final aligned video is rendered. We denote the coordinates and time stamps in the frame coordinate with plain letters, in the sequence coordinate with $\sim$ over the notation, e.g., $\tilde{\mathbf{x}}$, and in the world coordinate with double $\sim$, e.g., $\tilde{\tilde{\mathbf{x}}}$. Accordingly, a transformation from the frame to sequence coordinate has $\sim$ over the notation, and a transformation from the sequence to world coordinate has double $\sim$. We use superscript for the sequence number and subscript for, either the frame number or trajectory number. E.g., $\tilde{\mathbf{h}}_i^s$ is the transformation of frame i in sequence s from the frame coordinate to sequence coordinate.

3.1. Spatial alignment

We break down the spatial alignment to two phases. First, for each sequence, we map all the frames to the sequence coordinate, via global motion compensation (GMC), which also produces a reconstructed background mosaic (Fig. 3(a)). A crucial assumption for successful GMC is the camera having small motion in the optical axis direction. Second, image alignment is conducted on the reconstructed backgrounds and maps them to the world coordinate (Fig. 3(b)). However, if the backgrounds are non-overlapping, common image alignment cannot be used. Thus, a new alignment scheme is proposed in Sec. 3.1.2.

3.1.1 Global motion compensation

GMC removes intentional or unwanted camera motion in a sequence, creating a video with static background [26, 27]. Essentially, GMC estimates a per frame transformation to the sequence coordinate. We utilize the TRGMC algorithm [26] which handles dynamic scenes and estimates transformations by jointly aligning input frames. TRGMC detects SURF [5] keypoints in each frame, and matches keypoints to densely interconnect all frames, regardless of

their temporal offset. These connections are referred as *links*. Then, appropriate transformation is applied to each frame and its links, such that the spatial coordinates of the end-points of each link are as similar as possible. Using TRGMC independently for each sequence s, we estimate the mapping $\tilde{\mathbf{h}}_i^s$ to map frame i to the sequence coordinate.

TRGMC defines the problem as the congealing of densely connected keypoints in a stack of frames. For the convenience of readers, we briefly introduce this algorithm. Given a stack of frames with indices $i \in \mathbb{K} = \{k_1, ..., k_N\}$, TRGMC is formulated as an optimization problem,

$$\min_{\{\tilde{\mathbf{h}}_i^s\}} \epsilon^s = \sum_{i \in \mathbb{K}} [\mathbf{e}_i(\tilde{\mathbf{h}}_i^s)]^\intercal \Omega_i^s [\mathbf{e}_i(\tilde{\mathbf{h}}_i^s)], \tag{1}$$

where $\tilde{\mathbf{h}}_i^s$ is an 8-dim homography transformation parameter from frame i of sequence s to the sequence coordinate, $\mathbf{e}_i(\tilde{\mathbf{h}}_i^s)$ collects pair-wise alignment errors of frame i relative to all other frames, and Ω_i^s is a weight matrix. Since TRGMC uses homography transformation, it works best with nodal camera motion. In the case of camera translation, TRGMC still works by matching the dominant background, although the result may downgrade with parallax.

The alignment error of frame i relative to all other frames is the sum of squared differences (SSD) between the start and end coordinates of the links connected to frame i, denoted for the kth link as $(x_{i,k}, y_{i,k})$ and $(u_{i,k}, v_{i,k})$, respectively. Thus, the error $\mathbf{e}_i(\tilde{\mathbf{h}}_i^s)$ is,

$$\mathbf{e}_i(\tilde{\mathbf{h}}_i^s) = [\mathbf{\Delta x}_i(\tilde{\mathbf{h}}_i^s)^\intercal, \mathbf{\Delta y}_i(\tilde{\mathbf{h}}_i^s)^\intercal]^\intercal, \tag{2}$$

where $\mathbf{\Delta x}_i(\tilde{\mathbf{h}}_i^s) = \tilde{\mathbf{w}}_i^{(x)} - \mathbf{u}_i$ and $\mathbf{\Delta y}_i(\tilde{\mathbf{h}}_i^s) = \tilde{\mathbf{w}}_i^{(y)} - \mathbf{v}_i$ are the errors in x and $y-$axes. The vectors $\tilde{\mathbf{w}}_i^{(x)}$ and $\tilde{\mathbf{w}}_i^{(y)}$ denote the x and $y-$coordinates of $(x_{i,k}, y_{i,k})$ warped by the parameter $\tilde{\mathbf{h}}_i^s$, respectively.

Equation 1 is solved by taking the Taylor expansion around $\tilde{\mathbf{h}}_i^s$ and finding the increment $\Delta\tilde{\mathbf{h}}_i^s$ that minimizes,

$$\underset{\Delta\tilde{\mathbf{h}}_i^s}{\operatorname{argmin}} [\mathbf{e}_i(\tilde{\mathbf{h}}_i^s) + \frac{\partial \mathbf{e}_i(\tilde{\mathbf{h}}_i^s)}{\partial \tilde{\mathbf{h}}_i^s} \Delta\tilde{\mathbf{h}}_i^s]^\intercal \Omega_i^s [\mathbf{e}_i(\tilde{\mathbf{h}}_i^s) + \frac{\partial \mathbf{e}_i(\tilde{\mathbf{h}}_i^s)}{\partial \tilde{\mathbf{h}}_i^s} \Delta\tilde{\mathbf{h}}_i^s] + \gamma \Delta\tilde{\mathbf{h}}_i^{s\intercal} \mathcal{I} \Delta\tilde{\mathbf{h}}_i^s, \tag{3}$$

where $\Delta\tilde{\mathbf{h}}_i^{s\intercal} \mathcal{I} \Delta\tilde{\mathbf{h}}_i^s$ is a regularization term. The indicator matrix $\mathcal{I}$ is a diagonal matrix specifying which elements of $\Delta\tilde{\mathbf{h}}_i^s$ need a constraint. By setting the first-order derivative of Eqn. 3 to zero, a closed-form solution for $\Delta\tilde{\mathbf{h}}_i^s$ is obtained. $\tilde{\mathbf{h}}_i^s$ is estimated after enough iterations.

Spatial alignment of overlapping backgrounds Given the $\tilde{\mathbf{h}}_i^s$ for all the input videos, we follow [26] to reconstruct the backgrounds B^s for them. If there exists enough overlap between the backgrounds, common image alignment algorithms may be used. Specifically, we estimate the transformation $\tilde{\tilde{\mathbf{h}}}^s$ that maps the background of sequence s to the world coordinate, by matching SURF keypoints on background images via the vector field consensus algorithm [20]. In summary, a point with the homogeneous coordinate $(x, y, 1)$ in frame i of sequence s is mapped to the

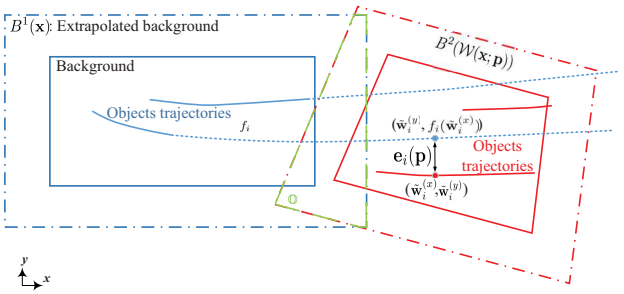

Figure 4. Spatial alignment of non-overlapping sequences using background extrapolation and smoothness of object trajectories.

sequence coordinate of sequence s, denoted as $(\tilde{x}, \tilde{y}, 1)$, and the world coordinate of all sequences, denoted as $(\tilde{\tilde{x}}, \tilde{\tilde{y}}, 1)$,

$$\left[\tilde{\tilde{x}}, \tilde{\tilde{y}}, 1\right]^\intercal = \tilde{\tilde{\mathbf{h}}}^s \left[\tilde{x}, \tilde{y}, 1\right]^\intercal = \tilde{\tilde{\mathbf{h}}}^s \tilde{\mathbf{h}}_i^s \left[x, y, 1\right]^\intercal. \tag{4}$$

Thus, the transformation $\tilde{\tilde{\mathbf{h}}}^s \tilde{\mathbf{h}}_i^s$ conducts spatial alignment for frame i in sequence s. Given the homography transformation of $\tilde{\tilde{\mathbf{h}}}^s$, as the cameras' baseline increases, the dominant background plane is aligned, and the foreground may be affected by parallax in the final composite video.

3.1.2 Spatial alignment of non-overlapping sequences

With freely panning cameras, it is likely that the backgrounds of sequences have no overlap, or the overall overlap is too small to reliably estimate the spatial alignment. One potential solution is to extrapolate the background images, and align the extrapolated images [23]. However, our experiments reveal that this is not reliable. First, extrapolation introduces many artifacts [3], or blurred areas [1, 23], leading to poor keypoint matching. Second, extrapolation in the horizontal direction, helps with alignment in the vertical direction, but leaves lots of ambiguity in horizontal alignment. Third, a rigid Euclidean transformation, as in [23], does not suffice for a proper background alignment.

On the other hand, how objects move across the sequences in the *spatial* world coordinate, irrespective of the temporal synchronization, provides hints for spatial alignment (Fig. 4). There is ambiguity in the exact spatial alignment, however, as more objects move across the sequences and in more diverse directions, the ambiguity is decreased.

To this end, we propose a spatial alignment algorithm for NOS that combines both aforementioned ideas. We first extrapolate the background images of all sequences. Then, we perform *motion tracking* to obtain trajectories of all keypoints in each sequence. By transforming the trajectories to the sequence coordinate using $\tilde{\mathbf{h}}_i^s$ and filtering out static trajectories, we collect moving object trajectories. We create *motion tracks* by matching moving object trajectories across sequences. Finally, we incrementally update the transformation applied to the background images to increase the motion track smoothness in the world coordinate,

while maintaining the appearance consistency of *extrapolated* backgrounds in the overlap region. In essence, instead of relying only on extrapolated appearance, which is blurry and unreliable, we base our method on the extrapolation that movement of objects provides across the FOV of sequences.

Motion tracking We perform tracking in consecutive frames to form the trajectories. We prefer *keypoint*-based tracking for two reasons. 1) Object-based tracking requires detecting generic objects on each frame, which could be error-prone and inefficient. 2) Our experiments and also the analysis in [17] reveal that optical flow-based tracking such as dense trajectories [30] leads to spurious motion trajectories around the motion boundaries. We use SURF [5] keypoint detection and description. To detect newly emerging objects, we start tracking all the keypoints on frame i that have no corresponding matches from frame $i-1$.

Denote the jth trajectory in sequence s as $P_j^s = [\mathbf{x}_j^s, \mathbf{y}_j^s, \tilde{\mathbf{t}}_j^s]$, where $\mathbf{x}_j^s$ and $\mathbf{y}_j^s$ are the frame coordinates, and $\tilde{\mathbf{t}}_j^s$ is the time stamp. To handle sequences at different frame rates, $\tilde{\mathbf{t}}_j^s$ should be the absolute time unit such as milliseconds not frame number. We then compute the trajectory $\tilde{P}_j^s$ in the sequence coordinate via $\tilde{\mathbf{h}}_i^s$. In this coordinate, trajectories of moving and stationary keypoints are easily distinguishable, as sequence coordinates of static objects remain constant over time (Fig. 3(d), bold vs. dashed lines). Denoting the trajectory length as l_j^s and width and height of the sequence as w^s and h^s, we omit stationary trajectories if,

$$\frac{1}{l_j^s}\sum_{k=1}^{l_j^s-1}\left(\frac{|\tilde{\mathbf{x}}_{j,k}^s - \tilde{\mathbf{x}}_{j,k+1}^s|}{w^s} + \frac{|\tilde{\mathbf{y}}_{j,k}^s - \tilde{\mathbf{y}}_{j,k+1}^s|}{h^s}\right) < \tau_1, \quad (5)$$

where τ_1 is a threshold for the total displacement of the tracked object, and $\tilde{\mathbf{x}}_{j,k}^s$ and $\tilde{\mathbf{y}}_{j,k}^s$ denote the kth element in the vectors $\tilde{\mathbf{x}}_j^s$ and $\tilde{\mathbf{y}}_j^s$, respectively.

Creating motion tracks We describe each moving object trajectory j of sequence s with two SURF descriptors, one for the keypoint starting the trajectory ($\mathcal{S}_j^s$) and one for the keypoint ending it ($\mathcal{E}_j^s$). To match two trajectories j and k from sequences s_1 and s_2, a classical keypoint matching algorithm [18] is used to match all 4 combinations of keypoints, i.e., $(\mathcal{S}_j^{s_1}, \mathcal{S}_k^{s_2})$, $(\mathcal{E}_j^{s_1}, \mathcal{E}_k^{s_2})$, $(\mathcal{S}_j^{s_1}, \mathcal{E}_k^{s_2})$ and $(\mathcal{E}_j^{s_1}, \mathcal{S}_k^{s_2})$, and the minimum distance decides a match. This can achieve more robustness against view point variation, as the nearby keypoints of the trajectories (in the world coordinate) will be the deciding factor in trajectory matching. We call each set of matched trajectories a *track*, denoted by Π_k. For simplicity of notation, we assume that the trajectories within a track have been re-indexed such that $\Pi_k = \{\tilde{\tilde{P}}_k^s; s \in [1, S]\}$. For a certain sequence s, $\tilde{\tilde{P}}_k^s$ might be empty, i.e., no trajectories from this sequence is part of the track Π_k. Note that not all the moving object trajectories should be matched to form tracks, due to noisy trajectories or objects with non-smooth motion path. Sec. 3.2.2 presents a method to remove non-smooth trajectories.

Spatial alignment formulation For simplicity, we discuss the alignment of 2 sequences, as more sequences may be aligned in the same manner, sequentially. We set $\tilde{\tilde{\mathbf{h}}}^1 = I_{3\times 3}$ and use $\mathbf{p}$ for $\tilde{\tilde{\mathbf{h}}}^2$ to avoid cluttered equations. Given N tracks indexed by i, and extrapolated backgrounds B^1 and B^2, the goal is to find a transformation $\mathbf{p}$ which maps B^2 to B^1, such that the extrapolated background are similar in the overlap region $\mathbb{O}(\mathbf{p})$ and trajectories of sequence 2 reside on the *extension* of trajectories in sequence 1, in $\tilde{\tilde{x}} - \tilde{\tilde{y}}$ coordinate. For image extrapolation, we use PatchMatch algorithm [3]. Then, we formulate the optimization problem (Fig. 4),

$$\min_{\mathbf{p}} \sum_{\tilde{\mathbf{x}}\in\mathbb{O}(\mathbf{p})} \left[B^2(\mathcal{W}(\tilde{\mathbf{x}};\mathbf{p})) - B^1(\tilde{\mathbf{x}})\right]^2 + \beta \sum_i \mathbf{e}_i(\mathbf{p})^\intercal \mathbf{e}_i(\mathbf{p}), \quad (6)$$

where $\mathcal{W}(\tilde{\mathbf{x}};\mathbf{p})$ warps $\tilde{\mathbf{x}}$ to the world coordinate by transformation $\mathbf{p}$, and $\mathbf{e}_i(\mathbf{p})$ represents how far trajectory i of sequence 2 is from spatial extension of matching trajectory in sequence 1. The first term in Eqn. 6 is similar to Lucas-Kanade algorithm [2], operated only in the overlapping area. To define $\mathbf{e}_i(\mathbf{p})$, we fit a line, which works better than fitting polynomials in our experiments, to the ith trajectory in sequence 1 (in the sequence coordinate), denoted by $f_i(x)$. The vector $\mathbf{e}_i(\mathbf{p})$ collects the y-distance between each point on the ith trajectory in sequence 2, after warped by $\mathbf{p}$, and the fitted line,

$$\mathbf{e}_i(\mathbf{p}) = [\tilde{\tilde{\mathbf{w}}}_i^{(y)} - f_i(\tilde{\tilde{\mathbf{w}}}_i^{(x)})], \quad (7)$$

where $\tilde{\tilde{\mathbf{w}}}_i^{(x)} = [\mathcal{W}_x(\tilde{x}_{i,2}, \tilde{y}_{i,2}; \mathbf{p})]$ and $\tilde{\tilde{\mathbf{w}}}_i^{(y)} = [\mathcal{W}_y(\tilde{x}_{i,2}, \tilde{y}_{i,2}; \mathbf{p})]$ are the warped $\tilde{x}$ and $\tilde{y}-$coordinates of the ith trajectory in sequence 2 to the world coordinate.

The optimization problem is solved by taking the Taylor expansion around $\mathbf{p}$ and finding the increment $\Delta\mathbf{p}$ by,

$$\begin{aligned}\underset{\Delta\mathbf{p}}{\operatorname{argmin}} &\sum_{\tilde{\mathbf{x}}\in\mathbb{O}(\mathbf{p})} \left[B^2(\mathcal{W}(\tilde{\mathbf{x}};\mathbf{p})) + S_B\Delta\mathbf{p} - B^1(\tilde{\mathbf{x}})\right]^2 \\ &+\beta\sum_i \left[\mathbf{e}_i(\mathbf{p}) + J_e\Delta\mathbf{p}\right]^\intercal \left[\mathbf{e}_i(\mathbf{p}) + J_e\Delta\mathbf{p}\right] + \alpha\Delta\mathbf{p}^\intercal\mathcal{I}\Delta\mathbf{p}, \end{aligned} \quad (8)$$

where $S_B = \nabla B^2 \frac{\partial\mathcal{W}}{\partial\mathbf{p}}$ is the steepest decent image, $J_e = \frac{\partial\mathbf{e}_i(\mathbf{p})}{\partial\mathbf{p}}$, and $\Delta\mathbf{p}^\intercal\mathcal{I}\Delta\mathbf{p}$ is a regularization term penalizing some special changes on $\Delta\mathbf{p}$ controlled by $\mathcal{I}$ and a positive constant α. By setting $\mathcal{I} = diag([0,0,1,0,0,1,0,0])$, we penalize large changes on translation elements of $\Delta\mathbf{p}$, so that frames are first aligned by warping them rather than translating them. Based on our experiments, this leads to more stable results. The solution to Eqn. 8 is,

$$\Delta\mathbf{p} = \mathbf{H}^{-1}\Big(\sum_{\tilde{\mathbf{x}}\in\mathbb{O}(\mathbf{p})} S_B^\intercal\left[B^1(\tilde{\mathbf{x}}) - B^2(\mathcal{W}(\tilde{\mathbf{x}};\mathbf{p}))\right] - \beta\sum_i J_e^\intercal\mathbf{e}_i(\mathbf{p})\Big), \quad (9)$$

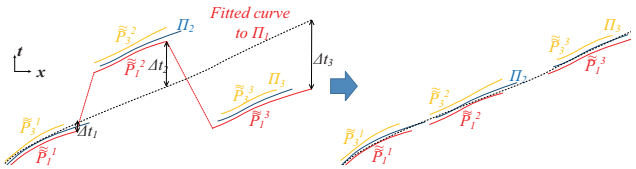

Figure 5. Trajectories, tracks, and fitted space-time curve to the tracks from three videos.

in which $\mathbf{H} = \sum_{\mathbf{x}\in\mathbb{O}(\mathbf{p})} S_B^\intercal S_B + \beta \sum_i J_e^\intercal J_e + \alpha\mathcal{I}$.

We initialize the algorithm by setting the sequences side by side (spatially) with the two possible layouts, and use the alignment result of the layout with lower final cost.

3.2. Temporal alignment

NOS are assumed to have moving objects, without which the temporal alignment is neither necessary nor possible. Given moving objects and spatial alignment results, the temporal alignment of NOS amounts to estimating *when* an object will appear in FOV of another camera, after it moves out of the current FOV. If both cameras observe the object's motion at the same time, the problem is easier. For this purpose, we create *motion tracks* as discussed in Sec. 3.1. Then, we *estimate the temporal offset* between sequences such that trajectories from the identical object follow a continuous path in $\tilde{\tilde{x}} - \tilde{\tilde{t}}$ and $\tilde{\tilde{y}} - \tilde{\tilde{t}}$ coordinates, i.e., the motion tracks are smooth. Since not all trajectories are due to moving objects, we *filter motion trajectories* with non-smooth paths, before matching trajectories.

3.2.1 Estimation of temporal offset

Given the collection of tracks, the objective is to make each track a smooth curve, by shifting the temporal coordinates of the contributing trajectories appropriately (Fig. 5). For S sequences, $\tilde{\tilde{x}}-$ coordinate of trajectories forming the kth track is the vector $[\tilde{\tilde{\mathbf{x}}}_k^1, \tilde{\tilde{\mathbf{x}}}_k^2, ..., \tilde{\tilde{\mathbf{x}}}_k^S]^\intercal$. $\tilde{\tilde{y}}-$ coordinate of each track is defined similarly. We assume that by temporally shifting each sequence s for Δt_s, the sequences are temporally aligned. To estimate Δt_s, we fit a polynomial curve of degree m to time stamps versus $\tilde{\tilde{x}}$ and $\tilde{\tilde{y}}-$coordinates of *each* track independently and estimate the time shifts, in order to achieve the lowest curve fitting error. Here, we discuss only the $\tilde{\tilde{t}} - \tilde{\tilde{x}}$ curve, and the $\tilde{\tilde{t}} - \tilde{\tilde{y}}$ curve is similar.

We denote the trajectory coordinates of sequence s and all the power terms of the polynomial space-time curve as

$$\tilde{\tilde{\mathbf{x}}}_k^{s(m)} = [\mathbf{1}_k^s, \tilde{\tilde{\mathbf{x}}}_k^s, [\tilde{\tilde{\mathbf{x}}}_k^s]^2, \cdots, [\tilde{\tilde{\mathbf{x}}}_k^s]^m], \tag{10}$$

where $\mathbf{1}_k^s$ is a l_k^s-dim vector of all ones, and $[.]^m$ denotes an element-wise power operation. For the track k, all the required terms of the polynomial space-time curve are collected in a matrix $\tilde{\tilde{X}}_k$ of size $\sum_s l_k^s \times (m+1)$, and all the time stamps in a vector $\tilde{\tilde{T}}_k(\Delta\mathbf{t})$ of length $\sum_s l_k^s$,

$$\tilde{\tilde{X}}_k = \begin{bmatrix} \tilde{\tilde{\mathbf{x}}}_k^{1(m)} \\ \tilde{\tilde{\mathbf{x}}}_k^{2(m)} \\ \vdots \\ \tilde{\tilde{\mathbf{x}}}_k^{S(m)} \end{bmatrix}, \tilde{\tilde{T}}_k(\Delta\mathbf{t}) = \begin{bmatrix} \tilde{\mathbf{t}}_k^1 + \Delta t_1 \\ \tilde{\mathbf{t}}_k^2 + \Delta t_2 \\ \vdots \\ \tilde{\mathbf{t}}_k^S + \Delta t_S \end{bmatrix}. \tag{11}$$

We denote the coefficients of the kth polynomial curve fitting to the kth track as $\mathbf{c}_k = [c_q]; q \in \{0, ..., m\}$. We can estimate the coefficients by solving a linear system, $\text{argmin}_{\mathbf{c}_k} \parallel \tilde{\tilde{T}}_k(\Delta\mathbf{t}) - \tilde{\tilde{X}}_k\mathbf{c}_k \parallel$. Since all tracks share the same $\Delta\mathbf{t}$, we can efficiently solve for all tracks jointly,

$$\mathbf{c}^*, \Delta\mathbf{t}^* = \underset{\mathbf{c},\Delta\mathbf{t}}{\text{argmin}} \parallel \tilde{\tilde{T}}(\Delta\mathbf{t}) - \tilde{\tilde{X}}\mathbf{c} \parallel, \tag{12}$$

in which

$$\tilde{\tilde{X}} = \begin{bmatrix} \tilde{\tilde{X}}_1 & 0 & \cdots & 0 \\ 0 & \tilde{\tilde{X}}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \tilde{\tilde{X}}_K \end{bmatrix}, \tilde{\tilde{T}}(\Delta\mathbf{t}) = \begin{bmatrix} \tilde{\tilde{T}}_1(\Delta\mathbf{t}) \\ \tilde{\tilde{T}}_2(\Delta\mathbf{t}) \\ \vdots \\ \tilde{\tilde{T}}_K(\Delta\mathbf{t}) \end{bmatrix}, \mathbf{c} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_K \end{bmatrix}. \tag{13}$$

Here, $\tilde{\tilde{X}}$ is a $N_K \times K(m+1)$ matrix where $N_K = \sum_k \sum_s l_k^s$ is the count of keypoints in all K tracks. We alternatively estimate $\mathbf{c}$ and $\Delta\mathbf{t}$, until the change in $\Delta\mathbf{t}$ is negligible. We first estimate $\mathbf{c}$, with fixed $\Delta\mathbf{t}$. Since $N_K \gg K(m+1)$, this linear system is over-constrained for $\mathbf{c}$. We solve $\mathbf{c}$ by Orthogonal-triangular decomposition, which is numerically more accurate than the pseudo inverse of $\tilde{\tilde{X}}$. Then, for a given $\mathbf{c}^*$, we set Δt_s as the average of residuals from the keypoints in trajectories belonging to sequence s,

$$\Delta t_s = -\frac{1}{N_s}(\tilde{\tilde{T}} - \tilde{\tilde{X}}\mathbf{c}^*)^\intercal \mathcal{I}_s, \tag{14}$$

where $\mathcal{I}_s$ is a binary indicator vector with an element equal to 1 if the corresponding row in $\tilde{\tilde{T}}$ comes from a trajectory in sequence s, and $N_s = \parallel \mathcal{I}_s \parallel_1$ is the count of such rows.

3.2.2 Motion trajectory filtering

As mentioned before, not all trajectories are resulted from object motion with a smooth path. In other words, some trajectories might be due to noise in keypoint locations while the camera moves. So, before matching trajectories across sequences and collecting them to a track, we filter out the trajectories that cannot be well approximated with a smooth path, by fitting the order-m polynomial to the trajectory,

$$\mathbf{c}_k^{s*} = \underset{\mathbf{c}_k^s}{\text{argmin}} \parallel \tilde{\mathbf{t}}_k^s - \tilde{\tilde{\mathbf{x}}}_k^{s(m)}\mathbf{c}_k^s \parallel_2, \tag{15}$$

and thresholding the total fitting residual to remove non-smooth trajectories, i.e., $\frac{1}{l_k^s} \parallel \tilde{\mathbf{t}}_k^s - \tilde{\tilde{\mathbf{x}}}_k^{s(m)}\mathbf{c}_k^{s*} \parallel_1 < \tau_2$.

4. Experimental results

In this section, we present the experimental setup and both quantitative and qualitative results. Note that since

Sequence ID	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	S1	S2	S3	S4	S5
Camera baseline (meter)	1	3	1	1	1	1	1	1	5	10	0	0	0	0	0
Temporal error (sec.)	0.13	0.07	0.07	0.13	0.07	0.07	0.10	0.03	0.07	0.07	0	0.03	0.07	0.03	0.07
Spatial error (pixel)	-	-	-	-	-	-	-	-	-	-	2	3	7	2	2

Table 1. Temporal and spatial alignment error in seconds and pixels, respectively, for real (R) and synthetic (S) sequences.

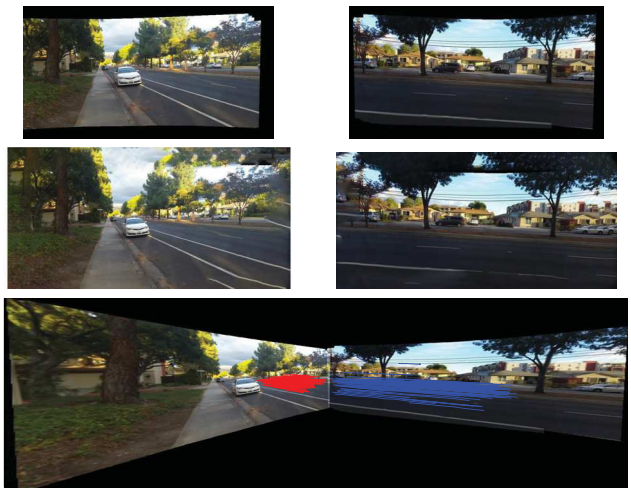

Figure 6. Spatial alignment of non-overlapping sequences. Top to bottom: reconstructed backgrounds of two sequences with negligible overlap, extrapolated backgrounds, and aligned background with trajectory of moving objects overlaid on the background.

NOS is a novel scenario for spatio-temporal alignment of sequences, there is no prior work for comparison. We set $\beta = 100$, $\alpha = 10^3$, $m = 3$ for the temporal curve fitting step, and $\tau_1 = 0.03$ and $\tau_2 = 0.15$ for trajectory filtering.

Dataset Given that there is no public dataset in this new scenario, we collect a NOS dataset including ten real-world sequence sets, and five synthetic sequence sets. Real sets are captured by two or three people using handheld smartphones with the distance between the cameras, i.e. baseline, as shown in Table 1. Synthetic sets provide sequences for which the ground truth result are exactly known, and are created by taking a sequence and cropping out two spatio-temporal tubes from the 3D sequence volume. This emulates the case of independently panning cameras with almost identical optical centers. To simulate a freely panning camera and hand shake, the spatial region used for each tube at each frame has a fixed size of 640×360 pixels, but the region location has an additive zero-mean Gaussian noise. Also, if the original video is stationary, the regions shift in $x-$direction to create a pan-like effect. The dataset is available at http://cvlab.cse.msu.edu/project-sequence-alignment.html.

Qualitative results Figure 6 presents the reconstructed backgrounds along with image extrapolation results. Further, it is shown how the backgrounds are transformed so that moving object trajectories have smooth path.

Figure 7 shows the alignment results for five sets of real sequences. The first two sets include sequences with some overall spatial overlap while the rest have no/minimal spatial overlap. For each set, two or three sample frames with moving objects are shown, at the time shift estimated by the proposed algorithm. Also, keypoint trajectories from both sequences in the world coordinate after spatio-temporal alignment are shown. Trajectories of moving objects have considerable extent in the $x-$direction, whereas trajectories of stationary objects are roughly parallel to $t-$axis. Finally, the two input frames are warped to the world coordinate to make a composite image. Although the input frames may not have direct overlap, perceived continuity of the scene and also relative location of the moving objects, demonstrate capabilities of the proposed algorithm and the application scenarios. Note that in all test sequences cameras move freely and independently, as shown by the range of trajectories in the world coordinate. For the case of "R7" in Fig. 7, the sequences are non-overlapping, but only a person is tracked moving across the FOVs. Thus, as shown in this figure, spatial alignment has some error, which consequently affects the accuracy of the temporal alignment.

Figure 8 represents a synthetic set where two sequences are created from a video of a car accident. The two cropped frames after spatio-temporal alignment are shown in a composite image and for comparison, the corresponding frame from the original video is also shown, demonstrating the accuracy of spatio-temporal alignment.

Quantitative results To quantitatively evaluate the proposed algorithm, we compare the alignment errors with the ground truth. For the case of synthetic sets, the original video from which the synthetic sequences are cropped, provides the ground truth location of the center points of the cropped frames. We measure the spatial location error of each aligned frame w.r.t. the ground truth location and report sum of absolute errors in x and $y-$direction, averaged over the sum of the length of the sequences, as the spatial alignment error. Also, since we create the synthetic sequences, the ground truth time shift is known. For real sets, when the input frames do not have overlap, quantifying the spatial error is not feasible. For quantification of temporal alignment, we manually align the sequences by relying on visual cues such as body pose, moving object location relative to background landmarks, and consistency of appearance of moving objects in the composite image. Table 1 provides the quantified temporal and spatial errors. As may be observed, temporal alignment works well even when the camera baseline distance increases, although the final consolidated result may suffer from parallax.

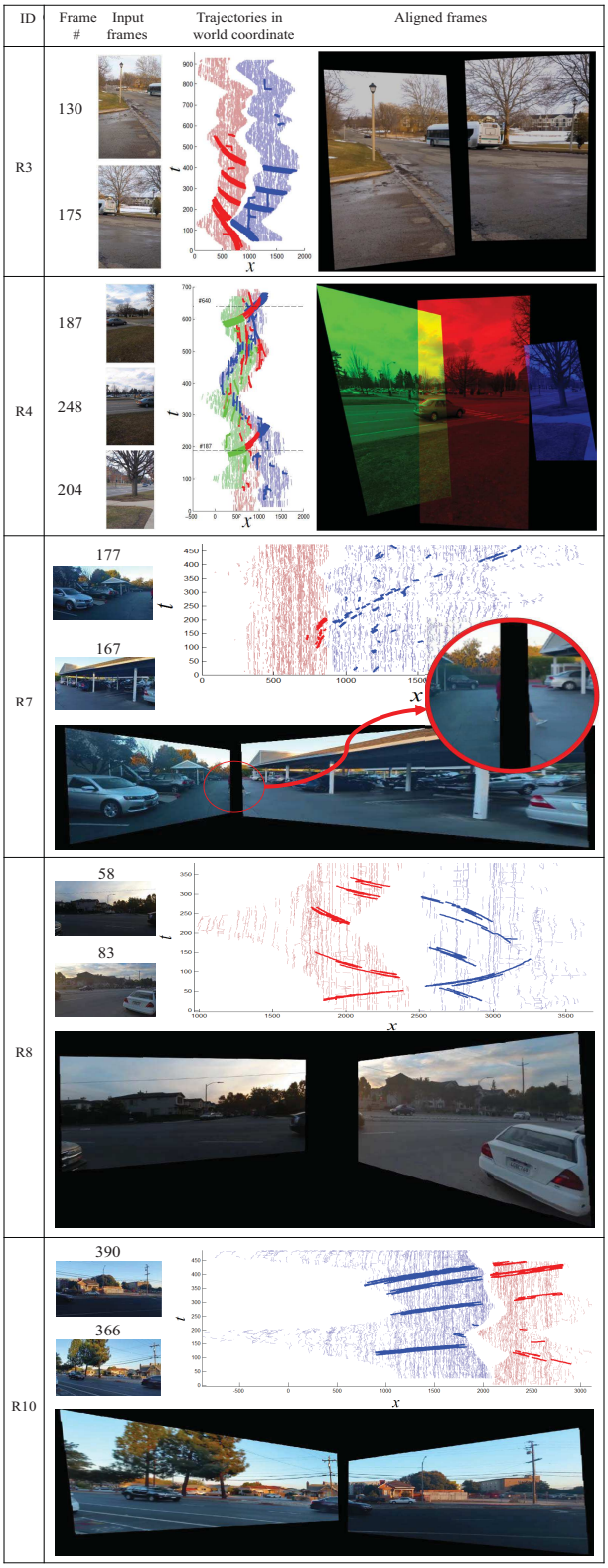

Figure 7. Each row shows spatio-temporal alignment results on a set of real NOS. For each sequence, input frames at the estimated time shift and trajectories of moving objects in the world coordinate are shown. The input frames are transformed to the world coordinate to make a composite image.

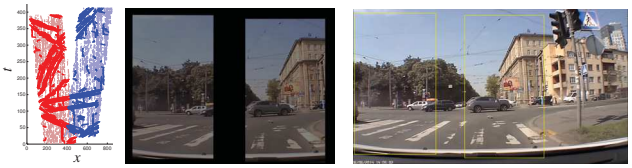

Figure 8. Results for two synthetic NOS from an accident footage (S1). Left to right: trajectories of moving objects, aligned input frames, original frame where the synthetic frames are cropped.

Computational cost The main computational cost of the proposed algorithm comes from TRGMC. On average, for a video of 15-second long, we spend 450 seconds on TRGMC and background reconstruction, using a PC with an Intel i5-3470@3.2GHz CPU, and 8GB RAM. Spatial alignment is independent of sequence length and takes $\sim$162 seconds on average for NOS. Finally, temporal alignment takes about 13 seconds on average over the dataset.

Limitations The proposed algorithm is a first step for generating panoramic videos in the challenging scenario of NOS. While this work relaxes many common assumptions of prior works, violation of some assumptions, especially existence of moving objects with a trajectory that spans FOVs of multiple cameras, results in alignment failures. Furthermore, when relying on non-rigid or articulated moving objects for alignment, many keypoints are not tracked long enough due to change of appearance, making alignment difficult. Also, in this case, matching trajectories among different sequences is less reliable. Since our algorithm is independent of the type of tracker, other tracking algorithms can be investigated in the future. Finally, alignment of non-overlapping background images suffers from ambiguity and is error prone, although the proposed algorithms makes use of available cues to conduct this task.

5. Conclusions

We proposed an algorithm for spatio-temporal alignment of sequences, referred to as non-overlapping sequences (NOS), from freely panning cameras whose FOVs might not even observe a common region over progression of time. This new scenario of video alignment is useful in reconstructing events, incidents, or crime scenes from multiple amateur-captured sequences, or creation of panoramic videos from cooperative users via handheld cameras. The spatial alignment of our algorithm relies on reconstructing background for each sequence and aligning the backgrounds. When backgrounds are non-overlapping, the spatial alignment uses clues from smoothness of moving objects' paths and coherent appearance of background after image extrapolation. Smoothness of trajectory of moving objects is also utilized as a clue for temporal alignment. Our experiments demonstrate capabilities of the proposed method, despite the challenging scenario of NOS.

Acknowledgement This work was partially supported by TechSmith Corporation.

References

[1] A. Aides, T. Avraham, and Y. Y. Schechner. Multiscale ultrawide foveated video extrapolation. In *Computational Photography (ICCP), 2011 IEEE International Conference on*, pages 1–8. IEEE, 2011. 4

[2] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004. 5

[3] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized patchmatch correspondence algorithm. In *European Conference on Computer Vision*, pages 29–43. Springer, 2010. 4, 5

[4] T. Basha, Y. Moses, and S. Avidan. Photo sequencing. In *European Conference on Computer Vision*, pages 654–667. Springer, 2012. 3

[5] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Proc. European Conf. Computer Vision (ECCV)*, pages 404–417. Springer, 2006. 3, 5

[6] D. N. Brito, F. L. Pádua, R. L. Carceroni, and G. A. Pereira. Synchronizing video cameras with non-overlapping fields of view. In *XXI Brazilian Symp. Computer Graphics and Image Processing*, pages 37–44. IEEE, 2008. 2

[7] Y. Caspi and M. Irani. Aligning non-overlapping sequences. *Int. J. Computer Vision*, 48(1):39–51, 2002. 1, 2

[8] F. Diego, D. Ponsa, J. Serrat, and A. M. López. Video alignment for change detection. *IEEE Trans. Image Proc.*, 20(7):1858–1869, 2011. 1

[9] F. Diego, J. Serrat, and A. M. López. Joint spatio-temporal alignment of sequences. *IEEE Trans. Multimedia*, 15(6):1377–1387, 2013. 1, 2

[10] S. Esquivel, F. Woelk, and R. Koch. Calibration of a multi-camera rig from non-overlapping views. In *Pattern Recognition*, pages 82–91. Springer, 2007. 2

[11] G. D. Evangelidis and C. Bauckhage. Efficient and robust alignment of unsynchronized video sequences. In *Pattern Recognition*, pages 286–295. Springer, 2011. 1, 2

[12] G. D. Evangelidis and C. Bauckhage. Efficient subframe video alignment using short descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(10):2371–2386, 2013. 2, 3

[13] T. Gaspar, P. Oliveira, and P. Favaro. Synchronization of two independently moving cameras without feature correspondences. In *Proc. European Conf. Computer Vision (ECCV)*, pages 189–204. Springer, 2014. 2

[14] N. Hasler, B. Rosenhahn, T. Thormahlen, M. Wand, J. Gall, and H.-P. Seidel. Markerless motion capture with unsynchronized moving cameras. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 224–231. IEEE, 2009. 1

[15] W. Jiang and J. Gu. Video stitching with spatial-temporal content-preserving warping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 42–48, 2015. 2

[16] H. Kong, J.-Y. Audibert, and J. Ponce. Detecting abandoned objects with a moving camera. *IEEE Trans. Image Process.*, 19(8):2201–2210, 2010. 1

[17] C. Liu, W. T. Freeman, and E. H. Adelson. Analysis of contour motions. In *Advances in Neural Information Processing Systems (NIPS)*, pages 913–920, 2006. 5

[18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Computer Vision*, 60(2):91–110, 2004. 5

[19] C. Lu and M. Mandal. A robust technique for motion-based video sequences temporal alignment. *IEEE Trans. Multimedia*, 15(1):70–82, 2013. 3

[20] J. Ma, J. Zhao, J. Tian, A. L. Yuille, and Z. Tu. Robust point matching via vector field consensus. *IEEE Trans. Image Process.*, 23(4):1706–1721, 2014. 4

[21] F. L. Pádua, R. L. Carceroni, G. A. Santos, and K. N. Kutulakos. Linear sequence-to-sequence alignment. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(2):304–320, 2010. 1, 2

[22] F. Perazzi, A. Sorkine-Hornung, H. Zimmer, P. Kaufmann, O. Wang, S. Watson, and M. Gross. Panoramic video from unstructured camera arrays. In *Computer Graphics Forum*, volume 34, pages 57–68. Wiley Online Library, 2015. 2

[23] Y. Poleg and S. Peleg. Alignment and mosaicing of non-overlapping images. In *Computational Photography (ICCP), 2012 IEEE International Conference on*, pages 1–8. IEEE, 2012. 4

[24] D. Pundik and Y. Moses. Video synchronization using temporal signals from epipolar lines. In *Proc. European Conf. Computer Vision (ECCV)*, pages 15–28. Springer, 2010. 1

[25] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *Int. J. Computer Vision*, 50(2):203–226, 2002. 1

[26] S. M. Safdarnejad, Y. Atoum, and X. Liu. Temporally robust global motion compensation by keypoint-based congealing. In *Proc. European Conf. Computer Vision (ECCV)*, pages 101–119. Springer, 2016. 3, 4

[27] S. M. Safdarnejad, X. Liu, and L. Udpa. Robust global motion compensation in presence of predominant foreground. In *Proc. British Machine Vision Conf. (BMVC)*, 2015. 3

[28] J. Serrat, F. Diego, F. Lumbreras, and J. M. Álvarez. Synchronization of video sequences from free-moving cameras. In *Pattern Recognition and Image Analysis*, pages 620–627. Springer, 2007. 1

[29] Y. Ukrainitz and M. Irani. *Aligning sequences and actions by maximizing space-time correlations*. Springer, 2006. 1

[30] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 3169–3176. IEEE, 2011. 5

[31] O. Wang, C. Schroers, H. Zimmer, M. Gross, and A. Sorkine-Hornung. Videosnapping: Interactive synchronization of multiple videos. *ACM Trans. on Graphics (TOG)*, 33(4):77, 2014. 1, 2

[32] F. Zhang and F. Liu. Parallax-tolerant image stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3262–3269, 2014. 2