JA-POLS: a Moving-camera Background Model via Joint Alignment and Partially-overlapping Local Subspaces

Irit Chelly
Ben-Gurion University
tohamy@post.bgu.ac.il

Vlad Winter
Ben-Gurion University
winterv@post.bgu.ac.il

Dor Litvak
Ben-Gurion University
dorlit@post.bgu.ac.il

David Rosen
Massachusetts Institute of Technology
dmrosen@mit.edu

Oren Freifeld
Ben-Gurion University
orenfr@cs.bgu.ac.il

Abstract

Background models are widely used in computer vision. While successful Static-camera Background (SCB) models exist, Moving-camera Background (MCB) models are limited. Seemingly, there is a straightforward solution: 1) align the video frames; 2) learn an SCB model; 3) warp either original or previously-unseen frames toward the model. This approach, however, has drawbacks, especially when the accumulative camera motion is large and/or the video is long. Here we propose a purely-2D unsupervised modular method that systematically eliminates those issues. First, to estimate warps in the original video, we solve a joint-alignment problem while leveraging a certifiably-correct initialization. Next, we learn both multiple partially-overlapping local subspaces and how to predict alignments. Lastly, in test time, we warp a previously-unseen frame, based on the prediction, and project it on a subset of those subspaces to obtain a background/foreground separation. We show the method handles even large scenes with a relatively-free camera motion (provided the camera-to-scene distance does not change much) and that it not only yields State-of-the-Art results on the original video but also generalizes gracefully to previously-unseen videos of the same scene. Our code is available at https://github.com/BGU-CS-VIL/JA-POLS.

1. Introduction

Background modeling is an important video-analysis tool with applications such as tracking and change detection. In the static-camera case, the problem has been solved successfully[42, 11, 5, 1, 19, 6]. Our focus is on the more challenging case, where the camera is moving. There, the success has been more modest, as existing methods are limited to highly-restricted motions (*e.g.*: translations; small motions; jitter) and/or a small *accumulative* motion across the video; moreover, they do not directly generalize to previously-unseen misaligned videos. This raises a natural question: why not employ the following seemingly-simple 3-step solution? 1) align all the video frames (via, *e.g.*, [4, 27, 22, 8, 9, 21]); 2) learn a Static-camera Background (SCB) model of the global scene from the aligned frames; 3) warp previously-unseen frames toward the SCB model and apply the latter to the former, where the warping is done using, *e.g.*, either classical tools (see [44] and references therein) or methods such as PoseNet [25]. Unfortunately, this logical approach, whose first two steps are exemplified by the clever PRPCA [30], suffers from severe drawbacks (related to, among other things, scalability and optimization challenges) that hinder its applicability, especially for large scenes (*i.e.*, a large accumulative camera motion) and/or long videos. A more popular

Figure 1: JA-POLS' example results. The foreground is visualized via the sum of the RGB squared residuals.

This work was partially supported by the Lynn and William Frankel Center for Computer Science at BGU.

alternative focuses on incremental model updates to perform background/foreground (BG/FG) separation of the next frame in the video stream. The lack-of-memory property of the approach, however, prevents an effective use of all the previously-acquired data (especially when the camera returns to regions covered earlier, possibly at new orientations/positions). We also note that existing methods, whether global (*e.g.*, PRPCA) or incremental, target BG/FG separation in the original video and lack a readily-available mechanism to do so for previously-unseen unaligned frames (unless it is the next frame right at the end of the original video). Our approach is different. Particularly, *we propose a novel method, self-coined JA-POLS (short for Joint Alignment and Partially-overlapping Local Subspaces), for unsupervised learning of a Moving-camera Background (MCB) model*; see Fig. 1. JA-POLS, a purely-2D modular method, allows for large camera motions (either accumulative ones or between consecutive frames) and provides a mechanism for warping previously-unseen frames toward the model. The model itself scales gracefully since rather than trying to capture the background of the entire scene using a single low-dimensional global ("panoramic-size") subspace, it consists of multiple smaller *Partially-overlapping Local Subspaces* (POLS). As we show, JA-POLS not only yields State-of-the-Art (SOTA) results on the original data but also generalizes to previously-unseen unaligned videos. **Our key contributions are as follows:** 1) a novel MCB method that allows for a substantial and relatively-free camera motion; 2) the alignment of the original frames is done jointly (not pairwise) and utilizes an efficient initialization with theoretical guarantees; 3) all the alignment-related computations are done in 2D directly from image measurements, obviating the need of an explicit 3D-scene reconstruction, of constructing a global panoramic image, and of camera calibration. 4) The POLS model overcomes the issues that prevent a single global model from handling large scenes and/or long videos; 5) unlike competing methods, which focus only on BG/FG separation in the original video and/or the next frame, JA-POLS also provides a mechanism for aligning frames taken from new videos (covering the same scene but taken at possibly-different times and from possibly-different camera poses).

2. Related Work

The global approach to MCB modeling starts with building a representation of the entire scene. This usually involves, as preprocessing, aligning the frames of the original video, thereby reducing the MCB problem to a typically-large SCB problem with missing data [30].

Image alignment. In [10, 30], homographies between consecutive frames are estimated, while [24] uses a multi-layer homography. Works such as [48, 29] generate an adaptive panoramic image, while [45] assumes a PTZ camera. Also related is video stabilization; *e.g.*, [14] finds an optimal steady-camera path using pairwise transformations between consecutive frames, while [28] minimizes a global cost based on the warped frames. Most of the works above assume a calibrated camera and/or a highly-restricted camera motion (*e.g.*, small motions or PTZ). Moreover, transformation estimation is usually done pairwise and sequentially; this is prone to accumulative errors as well as perspective distortions when the scene is wide. AutoStitch [4] uses bundle adjustment upon computing pairwise geometric matches. Available implementations of [4] do not scale well with the number of input frames and typically handle, at most, only a few hundreds of frames. Other alignment methods use 3D data, *e.g.*, the (non-visual) Simultaneous Localization and Mapping (SLAM) [31, 26] which estimates a model of the environment together with a dynamic camera pose. Posenet [25] is a neural net that estimates camera poses from images and uses ground-truth 3D poses in its training. Such methods rely on depth data (in some SLAM methods) and/or expensive 3D reconstruction procedures such as Structure-from-Motion [47] (as in, *e.g.*, [25]). Our method includes a regression net which is, conceptually, akin to PoseNet; the differences are that ours is purely 2D-based and that what it predicts are invertible affine transformations.

Background models. For already-aligned images, SCB models have been researched extensively. Earlier methods focused on pixelwise models [42, 18, 52]. Thurnhofer-Hemsi *et al*. [45] use a competitive-learning net that learns receptive fields in the panoramic scene. Another main approach, closer to ours, is learning a low-dimensional subspace. Principal Component Analysis (PCA) can be used but only if it can be assumed that the data contains neither foreground objects nor outliers. Otherwise, Robust PCA (RPCA) methods are preferred. The first RPCA in computer vision was proposed in [11]. Later, Candes *et al*. [5] and similar works [51, 16] used a "low-rank and sparse" ("$L+S$") data decomposition. The low-rank part represents the background while the sparse part models outliers. Unfortunately, all these models [11, 5, 51, 16] do not scale. A scalable RPCA was proposed in [17] based on Trimmed Grassmann Averages (TGA); see also [6]. Works such as [1, 19, 15] use $L+S$ decompositions within subspace tracking; despite the word "tracking", these methods, which focus on subspace updates, are more suitable for SCB than MCB models. A related MCB approach, t-GRASTA [20], relies on [34] and alternates between motion estimation and subspace learning. DECOLOR [50] is a similar

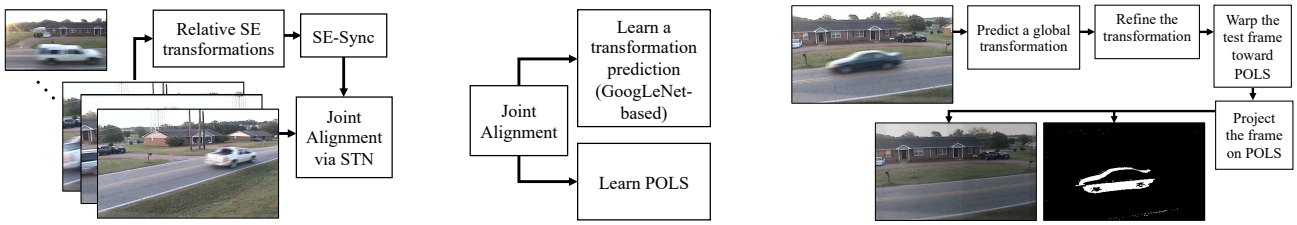


Figure 2: Flowcharts of JA-POLS: (1) the joint-alignment phase; (2) the learning phase; (3) the test phase.

MCB approach focusing on moving-object detection. These MCB methods linearly approximate the motions and thus cannot handle a large camera motion. The incPCP-PTI method [7] continuously estimates a rigid-body transformation between each new frame and the low-rank component of the previous one, and then applies it to the whole low-rank matrix. This method targets a PTZ camera. Several works focus on **moving-object detection in a moving camera**; *e.g.*, [49] segments optical flow into BG/FG while [38] classifies feature-based trajectories. These works, which solve a related but slightly-different problem from ours, cannot detect static changes and cannot handle long sequences; see also [2]. Lastly, all works mentioned in this section lack a direct way to handle previously-unseen misaligned frames (unless the frame is next consecutive one).

3. The Proposed Method: JA-POLS

The outline of the proposed method, which treats the original video as (unlabeled) *training data*, is as follows. **Phase 1 (§ 3.1):** Given a training video, we solve an unsupervised *joint-alignment* problem using: a novel smart initialization with theoretical guarantees; a Lie-algebraic parameterization/regularization; a Spatial Transformer Net (STN). See [41] for benefits of a Lie-algebraic parameterization (via the matrix exponential) of affine maps within STNs. **Phase 2 (§ 3.2):** Upon the joint alignment, we learn two tasks independently: 1) *alignment prediction*; 2) *learning multiple local low-dimensional robust background models over partially-overlapping areas*, yielding a set of local linear subspaces, each associated with a different region of the scene. **Phase 3 (§ 3.3):** In test time, a new frame is first warped toward the global scene via the (refined) predicted alignment, and then projected on the Partially-overlapping Local Subspaces (POLS). The average of the projections results in an effective BF/FG separation. See flowcharts in Fig. 2. Importantly, our method requires neither the creation of a single global model for an entire panoramic scene nor 3D reconstruction; rather, it employs a decentralized and localized approach and is purely 2D-based. As our method uses Lie groups/algebras, our **Sup. Mat.** contains all the relevant required background used below.

Notation. Let $\mathrm{SE}(2)$ and $\mathrm{Aff}(2)$ denote the Special Euclidean and affine groups in 2D, respectively. Both groups can be seen as nonlinear spaces of 3-by-3 matrices acting on $\mathbb{R}^2$ (in homogeneous coordinates) and $\mathrm{SE}(2) \subsetneq \mathrm{Aff}(2)$. Let $\mathfrak{aff}(2)$ denote the Lie algebra of $\mathrm{Aff}(2)$; $\mathfrak{aff}(2)$ is a 6D linear space of 3-by-3 matrices. Let $\mathbf{vec} : \mathfrak{aff}(2) \to \mathbb{R}^6$ denote a linear bijection. The matrix exponential and logarithm, $\exp : \mathfrak{aff}(2) \to \mathrm{Aff}(2)$ and $\log : \mathrm{Aff}(2) \to \mathfrak{aff}(2)$, connect the algebra to the group. If $\boldsymbol{\theta} \in \mathbb{R}^6$, then $T^{\boldsymbol{\theta}} = \exp(\mathbf{vec}^{-1}(\boldsymbol{\theta})) \in \mathrm{Aff}(2)$ is the affine transformation parameterized by $\boldsymbol{\theta}$, and $d(T^{\boldsymbol{\theta}}, \mathrm{SE}(2))$ (see **Sup. Mat.**) measures how far $T^{\boldsymbol{\theta}}$ is from $\mathrm{SE}(2)$. Let Ω_{scene} be the domain of the panoramic-size image, and let D be the number of its pixels.

3.1. Unsupervised Joint Alignment

Given training frames, $(x_i)_{i=1}^N$, we seek $(T^{\boldsymbol{\theta}_i})_{i=1}^N \subset \mathrm{Aff}(2)$ that minimize the (robustified) variance of all RGB values over the warped images, $(\widetilde{x}_i^{\boldsymbol{\theta}_i})_{i=1}^N$, where $\widetilde{x}_i^{\boldsymbol{\theta}_i} = x_i \circ T^{\boldsymbol{\theta}_i}$. Let $\widetilde{d}_i^{\boldsymbol{\theta}_i} < D$ be the number of pixels in $\widetilde{\Omega}_i^{\boldsymbol{\theta}_i}$, the domain of $\widetilde{x}_i^{\boldsymbol{\theta}_i}$. Note that $\widetilde{\Omega}_i^{\boldsymbol{\theta}_i} \subsetneq \Omega_{\text{scene}} = \bigcup_{i=1}^N \widetilde{\Omega}_i^{\boldsymbol{\theta}_i}$. Let w_l^i denote a binary weight indicating whether or not pixel l in Ω_{scene} is also in $\widetilde{\Omega}_i^{\boldsymbol{\theta}_i}$, and let $\widetilde{x}_{il}^{\boldsymbol{\theta}_i}$ denote pixel l of $\widetilde{x}_i^{\boldsymbol{\theta}_i}$ (if $w_l^i = 0$, then $\widetilde{x}_{il}^{\boldsymbol{\theta}_i}$ is undefined). Consider the following minimization of a robust *joint-alignment loss*,

$$\min_{(\boldsymbol{\theta}_i)_{i=1}^N} \sum_{l=1}^{D} \sum_{i=1}^{N} w_l^i \rho(\widetilde{x}_{il}^{\boldsymbol{\theta}_i} - \mu_l, \sigma)$$

$$\mu_l = \frac{\sum_{i=1}^N w_l^i \widetilde{x}_{il}^{\boldsymbol{\theta}_i}}{\sum_{i=1}^N w_l^i}, \quad T^{\boldsymbol{\theta}_i} = \exp(\overbrace{\mathbf{vec}^{-1}(\boldsymbol{\theta}_i)}^{\in \mathfrak{aff}(2)}) \in \mathrm{Aff}(2) \quad (1)$$

where $\boldsymbol{\theta}_i \in \mathbb{R}^6$, μ_l is the average of *pixel stack* l, $\{\widetilde{x}_{il}^{\boldsymbol{\theta}_i} : w_l^i = 1\}$, and $\rho(\cdot, \sigma)$ is Huber's loss [3] of parameter $\sigma > 0$. This loss is akin to those used by others for joint alignment. In our setting, however, that loss is specially-hard to minimize: since usually $\widetilde{d}_i^{\boldsymbol{\theta}_i} \ll D$, and

since the optimal $\widetilde{\Omega}_i^{\boldsymbol{\theta}_i}$ and $\widetilde{\Omega}_j^{\boldsymbol{\theta}_j}$ of two warped images, $\widetilde{x}_i^{\boldsymbol{\theta}_i}$ and $\widetilde{x}_j^{\boldsymbol{\theta}_j}$, may be far from each other (implying that the difference between $T^{\boldsymbol{\theta}_i}$ and $T^{\boldsymbol{\theta}_j}$ is large, and that at least one of them is far from the identity map), minimizing this loss is likely to yield either bad local-minima or, worse, bad trivial global minima (*e.g.*, shrinking all images to a point or creating no overlap between the images.); we will return to this issue in our ablation studies. A potential remedy is regularizing the transformations' size and/or the difference between consecutive transformations; however, in a large scene it is hard to determine the amount of such regularizations. All this motivates us to *propose a novel loss function, over residual transformations*:

$$\min_{(\boldsymbol{\delta}_i)_{i=1}^N} \left(\sum_{i=1}^{N} \sum_{l=1}^{D} w_l^i \rho(\widetilde{x}_{il}^{\boldsymbol{\theta}_i} - \mu_l, \sigma) \right) + \lambda \sum_{i=1}^{N} d(T^{\boldsymbol{\theta}_i}, \mathrm{SE}(2))$$
$$\mu_l = \frac{\sum_{i=1}^N w_l^i \widetilde{x}_{il}^{\boldsymbol{\theta}_i}}{\sum_{i=1}^N w_l^i}, \; T^{\boldsymbol{\theta}_i} = \overbrace{\exp(\underbrace{\mathbf{vec}^{-1}(\boldsymbol{\delta}_i)}_{\in \mathfrak{aff}(2)})}^{\in \mathrm{Aff}(2)} \overbrace{g_i}^{\in \mathrm{SE}(2)} \in \mathrm{Aff}(2) \tag{2}$$

where $(g_i)_{i=1}^N \subset \mathrm{SE}(2)$ are *known*, $(\boldsymbol{\delta}_i)_{i=1}^N \subset \mathbb{R}^6$ parameterize the sought-after *residual* affine warps, and $\lambda > 0$ controls a new regularization term penalizing the deviation of the affine $T^{\boldsymbol{\theta}_i}$ from $\mathrm{SE}(2)$. Here, $\boldsymbol{\theta}_i$ is implied by $\boldsymbol{\theta}_i = \mathbf{vec}(\log(T^{\boldsymbol{\theta}_i})) \in \mathbb{R}^6$ (where $\log(T^{\boldsymbol{\theta}_i}) \in \mathfrak{aff}(2)$). The $(g_i)_{i=1}^N$ in Eq. (2) may be viewed as an initialization. A question then arises: *how can we find good values for this initialization?* After all, the aforementioned difficulties hold even if, in Eq. (1), the transformations are restricted to $\mathrm{SE}(2)$. Fortunately, there is a way to not only provide such good values but also do it in an efficient and scalable way. Upon obtaining $(g_i)_{i=1}^N$, as discussed below, we minimize the loss in Eq. (2) via an STN [23] whose input images are $(x_i \circ g_i)_{i=1}^N$.

A certifiably-correct initialization. Let x_i and x_j denote two input images. Let $\widetilde{g}_{ij} \in \mathrm{SE}(2)$ be a *noisy estimate* of a *relative* SE transformation warping x_j toward x_i (obtaining $\widetilde{g}_{ij}$ is discussed later). We use such pairwise transformations to *jointly* align the images $(x_i)_{i=1}^N$, in a *global* coordinate system. Concretely, we wish to estimate $(g_i)_{i=1}^N \subset \mathrm{SE}(2)$ that are as consistent as possible with the noisy relative transformations; *i.e.*, we want to have $\widetilde{g}_{ij} \approx g_i^{-1} g_j$ for all $(i,j) \in \mathcal{E}$, where $\mathcal{E}$ is a known subset of $(1,\dots,N) \times (1,\dots,N)$. This leads to the following known nonconvex estimation problem over the $3N$-dimensional nonlinear space $\mathrm{SE}(2)^N$ [37].

Definition 1 (The SE-Synchronization problem) *Given* $(\widetilde{\boldsymbol{g}}_{ij})_{i,j} \subset \mathrm{SE}(2)$, *find*

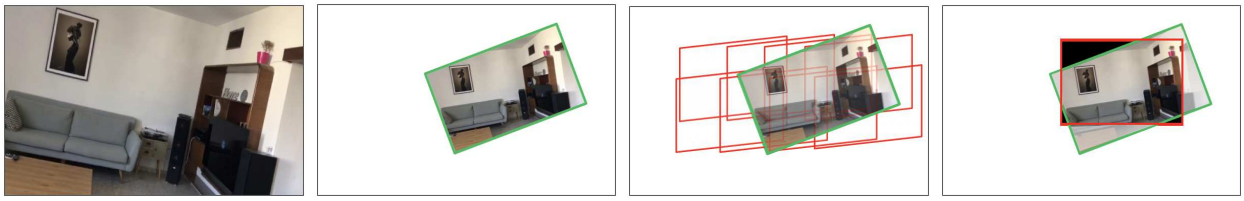
$$(g_i)_{i=1}^N = \underset{\boldsymbol{t}_i \in \mathbb{R}^2, \boldsymbol{R}_i \in \mathrm{SO}(2)}{\arg\min} \sum\nolimits_{(i,j)\in\mathcal{E}} \kappa_{ij} \|\boldsymbol{R}_j - \boldsymbol{R}_i \widetilde{\boldsymbol{R}}_{ij}\|_F^2 + \tau_{ij} \|\boldsymbol{t}_j - \boldsymbol{t}_i - \boldsymbol{R}_i \widetilde{\boldsymbol{t}}_{ij}\|_{\ell_2}^2 \tag{3}$$

where $\|\cdot\|_F^2$ *is the Frobenius norm,* $\widetilde{\boldsymbol{g}}_{ij} = \begin{bmatrix} \widetilde{\boldsymbol{R}}_{ij} & \widetilde{\boldsymbol{t}}_{ij} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \in \mathrm{SE}(2)$, $\kappa_{ij} > 0$, $\tau_{ij} > 0$, *and* $g_i = \begin{bmatrix} \boldsymbol{R}_i & \boldsymbol{t}_i \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \in \mathrm{SE}(2)$.

To solve it, we employ SE-Sync [37], an efficient and certifiably-correct algorithm for synchronization over $\mathrm{SE}(2)$ (more generally, $\mathrm{SE}(n)$). SE-Sync recovers *certifiably globally-optimal* solutions provided that the noise corrupting the $(\widetilde{g}_{ij})$ is not too large; moreover, even when exact optimality fails to hold, SE-Sync still produces a reasonable *approximated* solution, together with an *upper bound* on that solution's (global) suboptimality. SE-Sync thus provides us with a good estimate of $(g_i)_{i=1}^N \subset \mathrm{SE}(2)$ to jointly align $(x_i)_{i=1}^N$, where the alignments are restricted to $\mathrm{SE}(2)$. As mentioned above, we subsequently refine the transformations over the larger group, $\mathrm{Aff}(2)^N$, using the STN. An instance of SE-Sync is modeled as a sparse and nonlinear undirected graph $(\mathcal{V}, \mathcal{E})$; the nodes, $\mathcal{V}$, correspond to (the coordinate systems of the) input frames, x_i, while the edges, $\mathcal{E}$, correspond to a set of noisy estimates of *relative* transformations; *i.e.*, $\mathcal{E}_{ij}$ corresponds to $\widetilde{g}_{ij}$, the estimated transformation from coordinate system j to coordinate system i. We build $\mathcal{E}$ by connecting each node only with the next five ones; the rationale is that relative transformations between frames in such a short batch are usually small. The estimation of $\widetilde{g}_{ij} \in \mathrm{SE}(2)$, is done via established vision tools; see our **Sup. Mat.** For each (x_i, x_j) image pair, the result of that estimation procedure is not only the (estimated) relative transformation, $\widetilde{g}_{ij} = \begin{bmatrix} \widetilde{\boldsymbol{R}}_{ij} & \widetilde{\boldsymbol{t}}_{ij} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix}$, but also (estimated) precisions, τ_{ij} and κ_{ij}, of $\widetilde{\boldsymbol{t}}_{ij} \in \mathrm{SO}(2)$ and $\widetilde{\boldsymbol{R}}_{ij} \in \mathrm{SO}(2)$, respectively, used in Eq. (3).

3.2. Learning

POLS learning. Given the original frames and their estimated global affine transformations, $(x_i, T^{\boldsymbol{\theta}_i})_{i=1}^N$, a seemingly-straightforward approach is to use the warped images, $(\widetilde{x}_i^{\boldsymbol{\theta}_i})_{i=1}^N$, to learn a subspace-based SCB model (*e.g.*, some k-dimensional subspace such as PCA or one of its robust variants [11, 5, 17, 6]) whose domain is Ω_{scene}. The subspace would be represented by an orthogonal $D \times k$ matrix $\boldsymbol{V}_{\text{scene}}$. We note, however, that doing so for very large scenes: 1) can be very expensive as D can be huge; 2) requires learning a model where, in each example, *most of the data is missing*, as typically $\widetilde{d}_i^{\boldsymbol{\theta}_i} \ll D$; 3) requires k



(a) (b) (c) (d)

Figure 3: The POLS model. (a) A training frame, x_i. (b) Its warped version, $\widetilde{x}_i$, embedded in the larger scene. (c) A set of sliding-window regions, marked in red, that overlap with $\widetilde{x}_i$ in 85% of their pixels. Each region is associated with a local subspace, learned from such $\widetilde{x}_i$'s. (d): Projection of a warped frame (either train or test), marked by a green border, on a local subspace, whose domain is marked by a red border. Black areas indicate the missing values in that projection. Moreover, the projection ignores pixels within the green border that fall outside the red one.

large enough to capture the variability the dynamic background exhibits in the entire scene; 4) can have (for long videos) a prohibitively-large memory footprint. We thus argue that, especially for large scenes and/or long videos, there must be a better way: *if all we want is BG/FG separation in a small region (*i.e.*, the size of a single frame), why should we do everything at a whole-scene scale?* As an alternative, we propose learning portions of the scene individually. Particularly, we split Ω_{scene} into M partially-overlapping domains; see Fig. 3c. Let $m \in \{1, \ldots, M\}$. In our experiments, each such domain, Ω_m, is a 250-by-420 rectangle, obtained via a raster scan of 30-pixel horizontal/vertical strides (this determines M). See **Sup. Mat.** for the effect of using other window sizes. Let $n_m < N$ denote the number of warped training images whose domain overlaps Ω_m by more than 85%. We denote these images by $\{\widetilde{x}_q^{\boldsymbol{\theta}_q}\}_{q=1}^{n_m} \subset \{\widetilde{x}_i^{\boldsymbol{\theta}_i}\}_{i=1}^{N}$. Let $d_m = d$ denote the (constant) number of pixels in Ω_m. We form a $(3d)$-by-n_m local-data matrix, $\boldsymbol{Y}_m$, whose generic column, $\boldsymbol{y}_q \in \mathbb{R}^{3d}$, contains the RGB values of $\tilde{x}_q^{\boldsymbol{\theta}_q}$ in indices corresponding to overlapping pixels between Ω_m and $\widetilde{\Omega}_q^{\boldsymbol{\theta}_q}$ (the domain of $\tilde{x}_q^{\boldsymbol{\theta}_q}$), and NaNs (to indicate missing data) in indices corresponding to pixels that are in Ω_m but not in $\widetilde{\Omega}_q^{\boldsymbol{\theta}_q}$. We can now apply, to each $\boldsymbol{Y}_m$, any off-the-shelf method for linear dimensionality reduction suitable for background modeling. Let $\boldsymbol{V}_m$ denote the subspace learned from $\boldsymbol{Y}_m$. Note that: 1) each $\boldsymbol{Y}_m$ is *much* smaller, in either dimension, than an analogous global-data matrix, $\boldsymbol{Y}_{\text{scene}}$, that consists of all of the pixels in all of the warped frames (as well as *many* NaNs); 2) the $(\boldsymbol{V}_m)$'s can be learned in parallel (our distributed implementation exploits that); 3) the relative portion of missing data in $\boldsymbol{Y}_m$ is *much* smaller than that in $\boldsymbol{Y}_{\text{scene}}$. *To summarize, a POLS model is highly scalable and suffers less (than a global model) from missing data.*

Alignment learning. Given a test frame, we seek to align it w.r.t. the global scene. In theory, one can use any off-the-shelf tool for pairwise alignment. However, this would suffer from two issues. 1) One would have to explicitly reconstruct a panoramic-size image. Tools for creating panoramas do not scale to a very large scene (*e.g.*, thousands of frames). Moreover, committing to specific values of the panorama is error prone, and these mistakes can hurt alignment to it. 2) For a very large panorama, it is hard and/or time consuming to estimate the alignment using standard tools. Inspired by PoseNet [25], we prefer a deep-net approach. However, PoseNet's training relies on expensive and error-prone 3D reconstruction pipeline. In contrast, we propose a purely-2D approach that reuses the estimated training warps. Given the already-available training pairs, $(x_i, T^{\boldsymbol{\theta}_i})_{i=1}^{N}$, our regression net learns to map x_i to $\boldsymbol{\theta}_i \in \mathbb{R}^6 \cong \mathfrak{aff}(2)$, the (Lie-algebraic) parameter of an affine transformation in 2D. Learning transformation parameters between a pair of images directly has been done, *e.g.*, in [13, 32]. Here, however, we are interested in learning the transformation for each frame to a global coordinate system. As sometimes a training video might consists of only tens or hundreds of frames, a too-low number for training, we resort to standard approaches for handling data scarcity: 1) Transfer learning [39]: in training, we merely fine-tune a GoogLeNet [43] pre-trained on ImageNet [12]. 2) Data augmentation (*e.g.*, [40]): we generate synthesized frames by applying affine warps to the training data. For more details, see our **Sup. Mat.**

3.3. Test

In test time, the regression net takes an input image, x, and predicts $\boldsymbol{\theta} = \boldsymbol{\theta}(x)$. We warp x by $T^{\boldsymbol{\theta}}$ to produce a warped image in the global coordinate system, $x' = x \circ T^{\boldsymbol{\theta}}$. The predicted alignment is sometimes imperfect (in terms of pixel-level accuracy), but it still locates x very close to its desired destination. Thus, it is easy to refine the prediction as follows (the more expensive PoseNet would have also needed a refinement due to

similarly-imprecise results; see our **Sup. Mat.**). Let Ω' be the domain of x'. We select only the training pixel stacks that are either in Ω' or near it (*e.g.*, of distance < 10 pixels). We compute the average image of these pixel stacks. Then, we apply a standard tool for estimating the *small* residual transformation between x' and the average image (see **Sup. Mat**). Let $\widetilde{x}$ denote the result of applying the refined warp, and let $\widetilde{\Omega}$ denote its domain (Fig. 3d, green border). Among the local subspaces we use only those whose domain has some overlap with $\widetilde{\Omega}$. We project $\widetilde{x}$ on the subspace of each such model as follows. Let $m \in \{1, \ldots, M\}$ and let x_m be the following image, of domain Ω_m (Fig. 3d, red border): x_m coincides with $\widetilde{x}$ (Fig. 3d, green border) on $\Omega_m \cap \widetilde{\Omega}$, and takes zero values on $\Omega_m \setminus \widetilde{\Omega}$ (Fig. 3d, black pixels). Recall there are d pixels in Ω_m. Let $\boldsymbol{\mu}_m$ denote the (vectorized) mean image associated with the k-dimensional subspace $\boldsymbol{V}_m$ (some of the subspace methods we experimented with do not use a mean image; for these, just take $\boldsymbol{\mu}_m$ to be the zero vector). Let $\boldsymbol{b}_m$ denote the projection of $\boldsymbol{x}_m$, the vectorized version of x_m, on $\boldsymbol{V}_m$. It is defined by

$$\boldsymbol{b}_m = \boldsymbol{\mu}_m + \boldsymbol{V}_m \boldsymbol{\alpha}_m \tag{4}$$

$$\boldsymbol{\alpha}_m = \arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^k} \left\| \boldsymbol{W}_m^{\frac{1}{2}} (\boldsymbol{V}_m \boldsymbol{\alpha} - (\boldsymbol{x}_m - \boldsymbol{\mu}_m)) \right\|_{\ell_2}^2 \tag{5}$$

where $\boldsymbol{W}_m$ is a diagonal $3d \times 3d$ matrix whose diagonal elements are 1 if they correspond to pixels in $\widetilde{\Omega}$, and 0 otherwise. This is a linear weighted least-squares problem, so it has a standard closed-form solution. The background image of $\widetilde{x}$, according to model m, is given by b_m, the "unvectorized" version of $\boldsymbol{b}_m$. Next, we compute the pixelwise average of the background images and warp the result back to the domain of x. Letting b denote the resulting "unwarped" average image, b and $f \triangleq x - b$ are the BG/FG separation of x by the POLS model; see, *e.g.*, Fig. 1 and Fig. 2.

3.4. Camera-motion Types Handled by the Method

Recall that while we *initialize* the STN using SE(2), we end up working with Aff(2). Thus, while SE(2) cannot handle, *e.g.*, changes in scale, JA-POLS can handle *some* variations of those distances, *as long as these variations are not substantial enough to completely break the initialization*; this caveat is the main limitation of our method. The method has no problems with camera motions that are roughly parallel to the scene (*e.g.*, arbitrary in-plane rotations or vertical/lateral translations). Thus, it enables relatively-free camera movements, in the sense described above and, particularly, handles large accumulative motions (which none of the competing methods can handle) that are roughly parallel to the scene. Moreover, not relying on motion cues, in test videos JA-POLS can handle even fast motions of camera/objects. It also detects changes due to removal/displacement of static background objects. Empirically, perspective effects come into play only in very short ranges (*e.g.*, <2 meters). At least conceptually the affine STN could have been replaced with a homographic STN to handle such cases as well.

4. Results

We show, via qualitative and quantitative evaluations, that JA-POLS consistently yields SOTA results on moving-camera videos. Please also see the videos in the **Sup. Mat.** Whenever possible (see below), we compare against PRPCA [30], DECOLOR [50], Prac-ReProCS [15] and incPCP-PTI [7] in camera-motion datasets, and t-GRASTA [20] in camera-jitter datasets. We consider 9 moving-camera benchmark videos: Tennis, Swing, Stroller, Stunt, Flamingo, Hike (DAVIS dataset [35], [36]), Horse (Freiburg-Berkeley dataset [33]), Sidewalk and ContinuousPan (CDNet 2014 dataset [46]); these videos come with Ground-Truth (GT) FG masks. We also add our own 5 real videos, captured by us: Jitter, GardenShort, Kitchen, FastMotion and GardenWideScene. To enable qualitative evaluation on those, we inserted synthesized FG objects into each of them (see the **Sup. Mat.** for some visual examples). Together, the 14 videos cover a variety of camera-motion types: jitter (Sidewalk, Jitter), short videos with fairly-free motion (Tennis, Stroller, Swing, Flamingo, Hike, Horse, GardenShort), wide-scenes and/or long videos (Stunt, ContinuousPan, GardenWideScene), fast motion (FastMotion), and an indoor scene (Kitchen). Some of the sequences contain occlusions and illumination variations. To quantify FG estimation, we set a fixed threshold on the estimated FG, and compute the F_{measure} index defined by $F_{\text{measure}} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$ where precision and recall are derived from the GT and thresholded FG binary masks. *Each local subspace in our POLS model* was learned using either TGA [17] ($k = 5$; 60% trimming) or the denoising RPCA (Denoising-RPCA) which was proposed as part of the pipeline in [30]. We found that both of them outperformed [5], that in small datasets (< 80 frames) Denoising-RPCA was the best, and that in larger ones TGA was the winner; see **Sup. Mat.** for details. We evaluate all methods on the same training images. As our competitors lack a direct way to process new test videos, only JA-POLS can be evaluated on the latter. Table 1 compares the different methods (except t-GRASTA, which is not designed to handle most of the sequences there). PRPCA, which uses a global model, faces memory issues in wide scenes and could not run

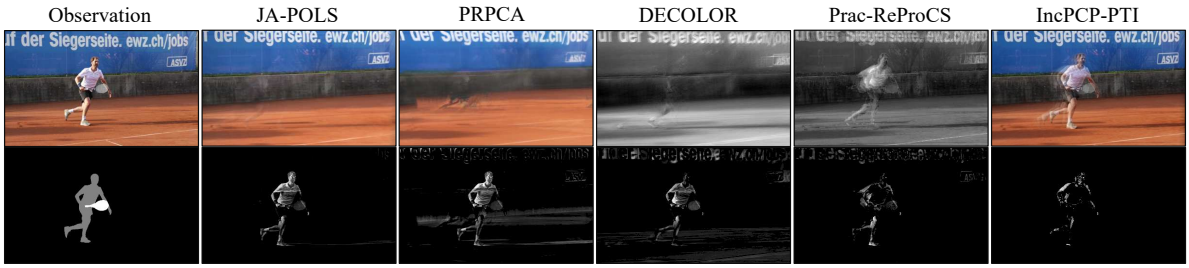


Figure 4: A typical frame from the Tennis dataset. Left column: observation (top row) and ground-truth foreground (bottom). Other columns: estimated background (top) and foreground (bottom) produced by each method.

Sequence	JA-POLS	PRPCA	DECOLOR	Prac-ReProCS	incPCP-PTI
Tennis	**0.67** $\pm$ 0.03	0.54 $\pm$ 0.03	0.21 $\pm$ 0.11	0.27 $\pm$ 0.03	0.16 $\pm$ 0.04
Swing	**0.62** $\pm$ 0.02	0.59 $\pm$ 0.05	0.27 $\pm$ 0.02	0.27 $\pm$ 0.01	0.19 $\pm$ 0.01
Stroller	**0.62** $\pm$ 0.03	0.50 $\pm$ 0.05	0.35 $\pm$ 0.04	0.21 $\pm$ 0.02	0.02 $\pm$ 0.01
Flamingo	**0.58** $\pm$ 0.01	0.50 $\pm$ 0.01	0.23 $\pm$ 0.01	0.14 $\pm$ 0.02	0.18 $\pm$ 0.03
Hike	**0.74** $\pm$ 0.01	0.70 $\pm$ 0.06	0.31 $\pm$ 0.02	0.07 $\pm$ 0.01	0.01 $\pm$ 0.00
Horse	**0.80** $\pm$ 0.01	0.61 $\pm$ 0.05	0.14 $\pm$ 0.03	0.27 $\pm$ 0.02	0.20 $\pm$ 0.04
Stunt	**0.48** $\pm$ 0.10	0.39 $\pm$ 0.11	0.10 $\pm$ 0.05	0.11 $\pm$ 0.05	0.15 $\pm$ 0.05
ContinuousPan	**0.67** $\pm$ 0.05	Out of memory	0.51 $\pm$ 0.08	0.59 $\pm$ 0.05	0.47 $\pm$ 0.09
Jitter	**0.83** $\pm$ 0.03	0.40 $\pm$ 0.05	0.22 $\pm$ 0.04	0.75 $\pm$ 0.04	0.53 $\pm$ 0.06
GardenShort	**0.83** $\pm$ 0.01	0.10 $\pm$ 0.00	0.38 $\pm$ 0.01	0.71 $\pm$ 0.01	0.34 $\pm$ 0.03
Kitchen	**0.57** $\pm$ 0.07	0.21 $\pm$ 0.02	0.19 $\pm$ 0.03	0.24 $\pm$ 0.04	0.24 $\pm$ 0.04
FastMotion	**0.41** $\pm$ 0.05	0.06 $\pm$ 0.01	0.11 $\pm$ 0.02	0.27 $\pm$ 0.03	0.21 $\pm$ 0.02
GardenWideScene	**0.44** $\pm$ 0.04	Out of memory	0.07 $\pm$ 0.02	0.35 $\pm$ 0.02	0.30 $\pm$ 0.03

Table 1: F-measure values (mean$\pm$std). The first 8 sequences are known benchmarks, the last 5 are our own.

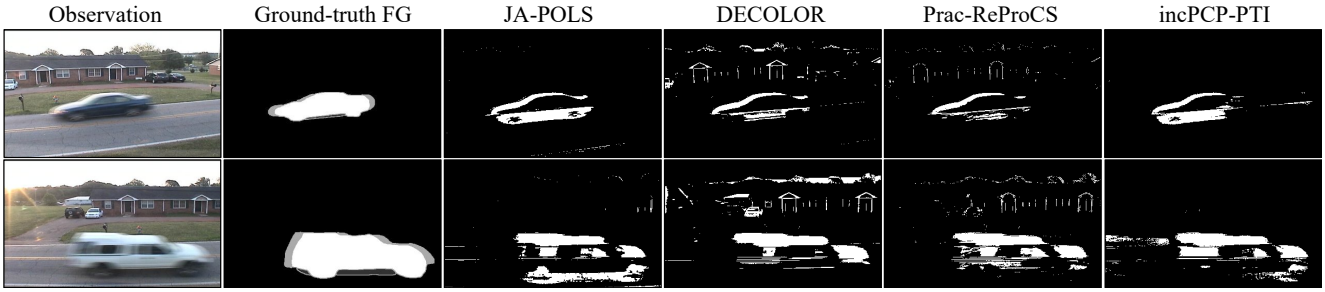


Figure 5: FG-extraction comparison of the methods on two typical frames from the ContinuousPan sequence.

Sequence	JA-POLS	t-GRASTA (batch)	t-GRASTA (online)
Sidewalk	**0.57** $\pm$ 0.05	0.45 $\pm$ 0.08	0.11 $\pm$ 0.07
Jitter	**0.83** $\pm$ 0.03	0.29 $\pm$ 0.03	0.44 $\pm$ 0.04
Kitchen	**0.57** $\pm$ 0.07	0.14 $\pm$ 0.01	0.18 $\pm$ 0.04

Table 2: F-measure performance (mean$\pm$std) of t-GRASTA and JA-POLS, on a small jitter case (Sidewalk), a medium jitter case (Jitter) and a relatively-free-motion case (Kitchen).

on them, even though the machine we used, whose full specs are at the **Sup. Mat.**, had 256GB RAM (to clarify, JA-POLS does not need so much RAM). Also, as PRPCA relies on pairwise alignments, it suffers from accumulative alignment errors, and these worsen as the video gets longer (*e.g.*, >100 frames). DECOLOR, Prac-ReProCS and incPCP-PTI degrade sharply when the motion is fast, suggesting they cannot handle rapid subspace evolutions. Figures 4 and 5 visualize typical BG/FG separation results produced by each method. For more visual and qualitative results, see **Sup. Mat.** Table 2 compares the performance, on the Jitter dataset, of JA-POLS and t-GRASTA [20] using 3 levels of motion. Figure 6 shows results for frames from a *test video*

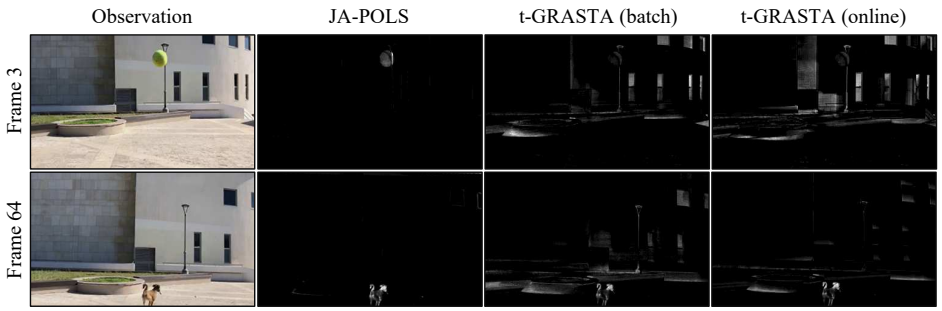


Figure 6: FG-extraction of JA-POLS vs. t-GRASTA (in its batch and online modes), on 2 typical frames from Jitter dataset. Note this is a *test video*. Since t-GRASTA cannot handle test videos, we ran it here from scratch (*i.e.*, not treating it as test data). While such a comparison is biased against JA-POLS, the latter still clearly wins. It also detects FG details even of small objects, and provides a relatively-clearer background.

Figure 7: Change detection: JA-POLS captures not only moving objects such as the man and the dog (which appears in the last 3 frames as can be seen upon zooming in) but also removal of static objects (a chair).

of the Jitter dataset (see the caption for details). Figure 7 shows frames from a test video of the GardenShort dataset. Note that a static object (a chair), which was part of the background, is moved to another location during the test video. JA-POLS detects that as it does not rely on motion cues.

JA-POLS evaluation on test videos. Leveraging the alignment predictor, we also evaluated JA-POLS' performance on 3 test videos. The resulting F-measure (mean±std) values are as follows: Kitchen: 0.50 ± 0.03, GardenWideScene: 0.55 ± 0.03 and Jitter: 0.88 ± 0.01.

Sequence	SE-Sync+STN		STN Only	
	F-measure	Loss	F-measure	Loss
Jitter	$\mathbf{0.83 \pm 0.03}$	**0.018**	0.80 ± 0.03	0.021
CP	$\mathbf{0.67 \pm 0.05}$	**0.064**	0.44 ± 0.07	0.123

Table 3: F-measure performance (mean±std) and the same STN loss (alignment+regularization) of JA-POLS on the Jitter and ContinuousPan (CP) data, with and without the SE-Sync initialization.

Ablation studies. Table 3 quantifies the importance of the SE-Sync initialization, showing its clear utility, especially when the motions go beyond mere jitters. The **Sup. Mat.** contains two additional such ablation studies. The first shows the importance of our regularization term, while the second shows that the alignment prediction is indeed indispensable.

Timings. SE-Sync takes a few seconds. POLS learning is fast, especially when TGA is used, and usually takes a few minutes. Since training the predictor is based on transfer learning, that too takes only minutes. The bottleneck is the STN optimization, whose running time, which ranges from ~ 15 minutes to several hours, depends on the length and complexity of the training video. However, processing a new test frame through the entire pipeline takes less than 2 [sec].

5. Conclusion

We proposed a novel MCB model and showed it achieves SOTA results and that it is highly scalable. We also showed that our choices in each step were judicious; *e.g.*, we demonstrated that POLS consistently outperforms a global model as well as the critical roles of the SE-Sync initialization, the novel regularization, and the predictor. While competing MCB models focus on BG/FG separation in the original data and/or incremental updates given the next frame, ours also generalizes to unseen misaligned videos (of the same scene, taken possibly at different times).

References

[1] Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. In *Allerton*, pages 704–711, 2010. 1, 2

[2] Matthew Berger and Lee M Seversky. Subspace tracking under dynamic dimensionality for online background subtraction. In *CVPR*, pages 1274–1281, 2014. 3

[3] Michael J Black and Anand Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *IJCV*, pages 57–91, 1996. 3

[4] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *IJCV*, pages 59–73, 2007. 1, 2

[5] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *JACM*, pages 1–37, 2011. 1, 2, 4, 6

[6] Rudrasis Chakraborty, Soren Hauberg, and Baba C Vemuri. Intrinsic grassmann averages for online linear and robust subspace learning. In *CVPR*, pages 6196–6204, 2017. 1, 2, 4

[7] Gustavo Chau and Paul Rodríguez. Panning and jitter invariant incremental principal component pursuit for video background modeling. In *ICCV*, pages 1844–1852, 2017. 3, 6

[8] Mark Cox, Sridha Sridharan, Simon Lucey, and Jeffrey Cohn. Least squares congealing for unsupervised alignment of images. In *CVPR*, pages 1–8, 2008. 1

[9] Mark Cox, Sridha Sridharan, Simon Lucey, and Jeffrey Cohn. Least-squares congealing for large numbers of images. In *ICCV*, pages 1949–1956, 2009. 1

[10] Carlos Cuevas, Raúl Mohedano, and Narciso García. Statistical moving object detection for mobile devices with camera. In *ICCE*, pages 15–16, 2015. 2

[11] Fernando De la Torre and Michael J Black. Robust principal component analysis for computer vision. In *ICCV*, pages 362–369, 2001. 1, 2, 4

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 5

[13] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv:1606.03798*, 2016. 5

[14] Matthias Grundmann, Vivek Kwatra, and Irfan Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *CVPR*, pages 225–232, 2011. 2

[15] Han Guo, Chenlu Qiu, and Namrata Vaswani. Practical reprocs for separating sparse and low-dimensional signal sequences from their sumpart 1. In *ICASSP*, pages 4161–4165, 2014. 2, 6

[16] Charles Guyon, Thierry Bouwmans, and El-Hadi Zahzah. Foreground detection via robust low rank matrix decomposition including spatio-temporal constraint. In *ACCV*, pages 315–320, 2012. 2

[17] Soren Hauberg, Aasa Feragen, and Michael J Black. Grassmann averages for scalable robust pca. In *CVPR*, pages 3810–3817, 2014. 2, 4, 6

[18] Eric Hayman and Jan-Olof Eklundh. Statistical background subtraction for a mobile observer. In *ICCV*, page 67, 2003. 2

[19] Jun He, Laura Balzano, and Arthur Szlam. Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. In *CVPR*, pages 1568–1575, 2012. 1, 2

[20] Jun He, Dejiao Zhang, Laura Balzano, and Tao Tao. Iterative grassmannian optimization for robust image alignment. *Image and Vision Computing*, pages 800–813, 2014. 2, 6, 7

[21] Gary Huang, Marwan Mattar, Honglak Lee, and Erik G Learned-Miller. Learning to align from scratch. In *NIPS*, pages 764–772, 2012. 1

[22] Gary B Huang, Vidit Jain, and Erik Learned-Miller. Unsupervised joint alignment of complex images. In *ICCV*, pages 1–8, 2007. 1

[23] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015. 4

[24] Yuxin Jin, Linmi Tao, Huijun Di, Naveed I Rao, and Guangyou Xu. Background modeling from a free-moving camera by multi-layer homography algorithm. In *ICIP*, pages 1572–1575, 2008. 2

[25] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *ICCV*, pages 2938–2946, 2015. 1, 2, 5

[26] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *International Symposium on Mixed and Augmented Reality*, pages 225–234, 2007. 2

[27] Erik G Learned-Miller. Data driven image models through continuous joint alignment. *TPAMI*, pages 236–250, 2006. 1

[28] Andrey Litvin, Janusz Konrad, and William Clement Karl. Probabilistic video stabilization using kalman filtering and mosaicing. In *Image and Video Communications and Processing*, pages 663–674. International Society for Optics and Photonics, 2003. 2

[29] Giulia Meneghetti, Martin Danelljan, Michael Felsberg, and Klas Nordberg. Image alignment for panorama stitching in sparsely structured environments. In *Scandinavian Conference on Image Analysis*, pages 428–439, 2015. 2

[30] Brian E Moore, Chen Gao, and Raj Rao Nadakuditi. Panoramic robust pca for foreground–background separation on noisy, free-motion camera video. *IEEE Transactions on Computational Imaging*, pages 195–211, 2019. 1, 2, 6

[31] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *ICCV*, pages 2320–2327, 2011. 2

[32] Ty Nguyen, Steven W Chen, Shreyas S Shivakumar, Camillo Jose Taylor, and Vijay Kumar. Unsupervised deep homography: A fast and robust homography estimation model. *Robotics and Automation Letters*, pages 2346–2353, 2018. 5

[33] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. *TPAMI*, 36(6):1187–1200, 2013. 6

[34] Yigang Peng, Arvind Ganesh, John Wright, Wenli Xu, and Yi Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *TPAMI*, pages 2233–2246, 2012. 2

[35] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016. 6

[36] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 6

[37] David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard. SE-Sync: A certifiably correct algorithm for synchronization over the special euclidean group. *The International Journal of Robotics Research*, 38(2-3):95–125, 2019. 4

[38] Yaser Sheikh, Omar Javed, and Takeo Kanade. Background subtraction for freely moving cameras. In *ICCV*, pages 1219–1225, 2009. 3

[39] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *TMI*, pages 1285–1298, 2016. 5

[40] Patrice Y Simard, Dave Steinkraus, and John C Platt. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, pages 958–963, 2003. 5

[41] Nicki Skafte Detlefsen, Oren Freifeld, and Søren Hauberg. Deep diffeomorphic transformer networks. In *CVPR*, pages 4403–4412, 2018. 3

[42] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, pages 246–252, 1999. 1, 2

[43] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 5

[44] Richard Szeliski. *Computer vision: algorithms and applications (Chapter 9)*. Springer Science & Business Media, 2010. 1

[45] Karl Thurnhofer-Hemsi, Ezequiel López-Rubio, Enrique Domínguez, Rafael Marcos Luque-Baena, and Miguel A Molina-Cabello. Panoramic background modeling for ptz cameras with competitive learning neural networks. In *IJCNN*, pages 396–403, 2017. 2

[46] Yi Wang, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar. Cdnet 2014: an expanded change detection benchmark dataset. In *CVPR Workshop*, pages 387–394, 2014. 6

[47] Changchang Wu. Towards linear-time incremental structure from motion. In *International Conference on 3D Vision*, pages 127–134, 2013. 2

[48] Kang Xue, Yue Liu, Jing Chen, and Qin Li. Panoramic background model for PTZ camera. In *International Congress on Image and Signal Processing*, pages 409–413, 2010. 2

[49] Hulya Yalcin, Martial Hebert, Robert Collins, and Michael J Black. A flow-based approach to vehicle detection and background mosaicking in airborne video. In *CVPR*, pages 1202–vol, 2005. 3

[50] Xiaowei Zhou, Can Yang, and Weichuan Yu. Moving object detection by detecting contiguous outliers in the low-rank representation. *TPAMI*, pages 597–610, 2012. 2, 6

[51] Zihan Zhou, Xiaodong Li, John Wright, Emmanuel Candes, and Yi Ma. Stable principal component pursuit. In *ISIT*, pages 1518–1522, 2010. 2

[52] Zoran Zivkovic and Ferdinand Van Der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, pages 773–780, 2006. 2