

A Deep Moving-camera Background Model

Guy Erez¹[0000-0002-4545-6664], Ron Shapira Weber¹[0000-0003-4579-0678], and Oren Freifeld¹[0000-0001-9816-9709]

Ben-Gurion University of the Negev, Be'er Sheva, Israel
{ergu, ronsha}@post.bgu.ac.il, orenfr@cs.bgu.ac.il

Abstract. In video analysis, background models have many applications such as background/foreground separation, change detection, anomaly detection, tracking, and more. However, while learning such a model in a video captured by a static camera is a fairly-solved task, in the case of a Moving-camera Background Model (MCBM), the success has been far more modest due to algorithmic and scalability challenges that arise due to the camera motion. Thus, existing MCBMs are limited in their scope and their supported camera-motion types. These hurdles also impeded the employment, in this unsupervised task, of end-to-end solutions based on deep learning (DL). Moreover, existing MCBMs usually model the background either on the domain of a typically-large panoramic image or in an online fashion. Unfortunately, the former creates several problems, including poor scalability, while the latter prevents the recognition and leveraging of cases where the camera revisits previously-seen parts of the scene. This paper proposes a new method, called DeepMCBM, that eliminates all the aforementioned issues and achieves state-of-the-art results. Concretely, first we identify the difficulties associated with joint alignment of video frames in general and in a DL setting in particular. Next, we propose a new strategy for joint alignment that lets us use a spatial transformer net with neither a regularization nor any form of specialized (and non-differentiable) initialization. Coupled with an autoencoder conditioned on unwarped robust central moments (obtained from the joint alignment), this yields an end-to-end regularization-free MCBM that supports a broad range of camera motions and scales gracefully. We demonstrate DeepMCBM's utility on a variety of videos, including ones beyond the scope of other methods. Our code is available at <https://github.com/BGU-CS-VIL/DeepMCBM>.

Keywords: unsupervised; background model; background subtraction; moving camera; joint alignment; regularization-free; deep learning; video analysis.

1 Introduction

The unsupervised video-analysis task this paper focuses on is learning a background model in a video captured by a moving camera. In the simpler case where the camera is static, such models have been used successfully in many computer-vision applications such as background/foreground separation, change or anomaly detection, and tracking. Static-camera solutions, however, cannot be easily extended to the moving-camera case since we do not know, a-priori, how the video frames should be aligned to each



(a) Examples for several input frames



(b) Alignment, visualized via the mean panoramic image (computed from the entire video)



(c) Background estimation using the Conditional Autoencoder

Fig. 1: Typical results of the proposed module. Note that despite the fact that the dog spent long times being static in two locations (as is evident by the corresponding ghosting effects in (b)) the model succeeded in eliminating it from the background.

other. Thus, most of the tools traditionally used in background models become less applicable; *e.g.*, methods based on learning a low-dimensional subspace via Robust Principal Component Analysis (RPCA) assume that the frames are aligned to each other.

Seemingly, there is a straightforward solution: “simply” align the frames to each other to reduce the problem back to the static-camera case, and then build a static-camera background model based on the aligned frames. However, this is more complicated than it might seem. First, the alignment problem itself is often difficult. For example, methods based on creating a panoramic image by sequentially aligning each pair of consecutive frames suffer from drift errors. Moreover, such methods cannot exploit the information conveyed in situations where the camera revisits (possibly from a different viewpoint) a previously-seen region in the scene. This, among other considerations, motivates solutions based on Joint Alignment (JA) of the frames. However, even in this formulation the problem is often still hard to solve, partially due to reasons we analyze later in § 4. Second, and regardless of how the alignment is done, there is the issue of scalability which pertains to not only the alignment problem itself but also the subsequent learning of the background model: when the accumulative motion of the camera throughout the video is substantial, the domain of the panoramic image can be huge so background models learned in that domain must scale gracefully. Furthermore, in such cases, when a frame is warped (*i.e.*, aligned) towards the panorama, it captures only a small portion of the latter. This means that *most* of the data in the panoramic version of the warped images is missing. This is problematic in our context since existing

solutions for subspace learning in the presence of missing data usually struggle in such cases. Therefore, the missing-data issue, together with the scalability requirement, considerably complicates the task. Due to the above reasons, the success in the case of a **Moving-camera Background Model (MCBM)** is lagging far behind its static-camera counterpart. Moreover, the difficulties above have also largely prevented the use of Deep Learning (DL) for this task. This is unfortunate not only because the idea of harnessing the power of DL is attractive but also since it hinders the usage of MCBMs within larger end-to-end pipelines.

With this in mind, the goal of this paper is to provide an effective and scalable DL-based MCBM. To that aim, we start by identifying more precisely what makes JA of video frames challenging: first in the general case and then in the more specific DL context. Next, we design a new JA strategy based on a regularization-free Spatial Transformer Net (STN) and a JA loss involving a memory aspect. Our method requires no auxiliary tools (such as the brittle and non-differentiable initialization used in [10]) that would prohibit its usage within end-to-end pipelines. We also propose a new deep module for learning a background model. The model, based on a Conditional Autoencoder (CAE) and the output of the JA module, is learned in the small domain of the input frames instead of the much-larger panoramic domain. This eliminates scalability issues and targets the goal of estimating the background more directly. Importantly, this module too can be used within end-to-end pipelines. [Figure 1](#) demonstrates the type of results obtained by the proposed modules. Taken together, the proposed two modules give rise to a new and highly-effective MCBM method, coined **DeepMCBM**, which supports a broad range of camera motions and scales gracefully. We demonstrate DeepMCBM’s utility on a variety of videos, including ones beyond the scope of competing methods.

Our key contributions are: 1) a DL module, for jointly aligning video frames, that relies on an STN-based optimization and a new training strategy that requires neither regularization nor initialization; 2) a DL background-modeling module that leverages the JA via a CAE conditioned on unwarped robust central moments derived from the JA; 3) together, these two modules form an end-to-end unsupervised MCBM that achieves SOTA results, that scales gracefully, and that supports a wide range of camera motions.

2 Related Work

STN [26] is a DL module that learns and applies a parameterized input-dependent spatial transformation. Given a parameterized transformation family and an input image f , the STN’s output consists of a parameter vector θ and a warped image obtained by warping f using T^θ (a transformation parameterized by θ). During training, the differentiation of a loss propagates through the STN. In practice, however, and despite their elegance, potential strength, and usage in numerous papers, STNs are often hard to train. Part of our solution addresses exactly such a case, where we take an STN-based optimization problem that was thought to be too difficult [10] and show how it can, in fact, be solved easily, without resorting to a regularization or a sophisticated limiting initialization.

Static-camera background models. Early methods were pixelwise (*e.g.*, [41]) but later the focus has shifted to subspace estimation using Robust Principal Component Analysis and its variants (*e.g.*, [44,7,50,20]). While those models usually do not scale

well, there also exist scalable RPCA models (*e.g.*, [21,8]).

Image alignment. In [13,34], pairwise homographies are estimated between consecutive frames while [27] uses a multi-layer homography. An adaptive panoramic image is built in [47,32] while [43] relies on the assumption that a PTZ camera is used. Most of the works above make stringent assumptions about the camera motion and estimate transformations between pairs of images, sometimes even sequentially. This approach, however, can lead to accumulative errors and/or significant distortions. To avoid such issues, AutoStitch [6] employs bundle adjustment. However, publicly-available implementations of AutoStitch scale poorly with the number of images (*e.g.*, cannot handle more than a few hundreds of frames). This is unlike the proposed approach which scales gracefully. Alignment methods relying on depth or expensive 3D information/reconstruction include [35,30,29,46]. Unlike those works, and similarly to, *e.g.*, [10], the JA approach in this paper is purely 2D-based.

MCBMs. Online RPCA methods (*e.g.*, [3,22,19]) were extended to the case of camera jitter [23] as well as more significant motions [18]. DECOLOR [49] is another MCBM, based on motion detection, that is restricted to small motions. IncPCP-PTI [9] targets a PTZ-camera setting by updating a low-dimensional subspace with the help of an estimated rigid motion between consecutive frames. Several MCBMs are built by first aligning the frames to each other, and then, in the usually-large domain of the obtained panoramic image, learning a background model from the warped images using a static-camera background model that can handle missing data (since each warped image covers only a portion of the panoramic domain). A prime example for such methods is PRPCA [34]. Also of note are methods targeting **moving-object detection in a moving camera**; *e.g.*, [48,39,4]. These works, however, cannot detect changes unrelated to motion and also do not scale well.

STN-based JA. As we explain in § 4, STN-based JA poses several difficulties. On that note, the closest work to ours is JA-POLS [10] which handles some of the difficulties via the usage of a non-differentiable and non-robust initialization, together with a fairly-restrictive regularization. While JA-POLS is effective in cases where it is applicable, it is limited in the camera-motion types it supports and is not an end-to-end solution. We will return to JA-POLS in more detail later on.

Learning background models in the panoramic domain. Once alignment is obtained, in principle a background model can be learned. However, panoramic-size models (*e.g.*, [34]) do not scale while using an ensemble of Partially-overlapping Local Subspaces (POLS) [10] is cumbersome and also suffers from the fact the number of models grows with the size of the panorama. Either way, the existing methods do not offer an end-to-end solution that can be used easily within DL pipelines.

3 Preliminaries: Joint Alignment (JA)

Let $(f^n)_{n=1}^N$ be the frames of the input video and assume the size of each frame is $h \times w$ pixels. Let C be the number of input channels; *e.g.*, $C = 3$ for RGB images (the case considered in this paper). Let $\Omega \subset \mathbb{R}^2$ denote the rectangular $h \times w$ common domain of each f^n , and let θ_n denote the (latent) parameter vector of the spatial transformation associated with the sought-after alignment of $f^n : \Omega \rightarrow \mathbb{R}^C$. The transformation itself,

denoted by T^{θ_n} , is viewed as an $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ map (not just $\Omega \rightarrow \mathbb{R}^2$). The value of $d = \dim(\theta_n)$ depends on the transformation family; *e.g.*, in the affine case, $d = 6$. The warped version of Ω is $\Omega^n \triangleq T^{\theta_n}(\Omega) \triangleq \{\mathbf{x} : \exists \mathbf{x}' \in \Omega \text{ s.t. } T^{\theta_n}(\mathbf{x}') = \mathbf{x}\} \subset \mathbb{R}^2$. Mathematically, we define the warped image as $g^n : \Omega^n \rightarrow \mathbb{R}^C$ using the equality

$$g^n(T^{\theta_n}(\mathbf{x}')) = f^n(\mathbf{x}') \quad \forall \mathbf{x}' \in \Omega. \quad (1)$$

However, due to technical reasons related to image warping [42], it is more convenient and customary to define g^n via the inverse transformation of T^{θ_n} :

$$g_{\mathbf{x}}^n \triangleq g^n(\mathbf{x}) = f^n((T^{\theta_n})^{-1}(\mathbf{x})) \quad \forall \mathbf{x} \in \Omega^n. \quad (2)$$

Note that g^n depends on θ_n and f^n . Let H and W be the height and width, respectively, of a rectangle, denoted by $\Omega_{\text{scene}} \subset \mathbb{R}^2$, that is large enough to contain $\bigcup_n \Omega^n$. We now define a mask that will be useful for reasons to become clear shortly. Let M^Ω be a single-channel $h \times w$ image whose domain is Ω and whose values are all equal to 1. Let $M^n : \Omega_{\text{scene}} \mapsto [0, 1]$ be a non-binary $H \times W$ mask obtained by image warping of M^Ω , according to T^{θ_n} , using zero padding and a bilinear interpolation kernel. That is, for any integral location \mathbf{x} in Ω_{scene} , the value of M^n at \mathbf{x} , denoted by $M_{\mathbf{x}}^n$, is given by

$$M_{\mathbf{x}}^n = \widetilde{M}_{\mathbf{x}'}^\Omega \quad \mathbf{x}' = T^{-\theta_n}(\mathbf{x}) \in \mathbb{R}^2 \quad (3)$$

where $\widetilde{M}_{\mathbf{x}'}^\Omega$ is interpolated from the values of M^Ω at the 4 integral locations nearest to \mathbf{x}' where whenever any of those integral locations falls outside Ω the value of M^Ω at that location is taken to be zero. Thus, $M_{\mathbf{x}}^n = 0$ if all those 4 locations are outside Ω , $M_{\mathbf{x}}^n = 1$ if they all fall inside it, and $0 < M_{\mathbf{x}}^n < 1$ otherwise. Let $g_{\mathbf{x},c}^n$ denote the value of $g_{\mathbf{x}}^n$ at channel c . We will refer to $p_{\mathbf{x},c} \triangleq (g_{\mathbf{x},c}^n)_{n=1}^N$ where $c \in \{1, \dots, C\}$ as the C pixel stacks at location \mathbf{x} . Similarly, we define the mask stack at location \mathbf{x} as $m_{\mathbf{x}} \triangleq (M_{\mathbf{x}}^n)_{n=1}^N$. Note that $p_{\mathbf{x},c}$ and $m_{\mathbf{x}}$ depend on $(\theta_n)_{n=1}^N$. A **joint-alignment loss**, to be minimized w.r.t. $(\theta_n)_{n=1}^N$, may be formulated in terms of

$$\mathcal{L}_{JA} = \text{func}((p_{\mathbf{x},c})_{c=1}^C, m_{\mathbf{x}})_{\mathbf{x} \in \Omega_{\text{scene}}}. \quad (4)$$

For example, in the early works on *congealing* (*e.g.*, [33,31,25,24]) that loss was based on entropy minimization. Later, other researchers [11,12] showed the benefits of a loss based on least squares. A robust variant (used in [10]) of the latter is

$$\mathcal{L}_{JA} = \frac{1}{N} \sum_{n=1}^N \frac{1}{C} \sum_{c=1}^C \frac{\sum_{\mathbf{x} \in \Omega_{\text{scene}}} M_{\mathbf{x}}^n \rho_{JA}(g_{\mathbf{x},c}^n - \mu_{\mathbf{x},c})}{\sum_{\mathbf{x} \in \Omega_{\text{scene}}} M_{\mathbf{x}}^n} \quad (5)$$

where $\mu_{\mathbf{x},c} = \frac{\sum_{n=1}^N M_{\mathbf{x}}^n g_{\mathbf{x},c}^n}{\sum_{n=1}^N M_{\mathbf{x}}^n}$ and ρ_{JA} is a differentiable robust error function [5].

Let μ be the mean of the warped images; *i.e.*, the value of μ at location \mathbf{x} and channel c is $\mu_{\mathbf{x},c}$. Note that μ may be viewed as the ‘‘moving target’’ to which the frames should be aligned. It ‘‘moves’’, during the optimization, in the following sense. As the alignment of the frames keeps changing, μ changes too since it is computed using the (weighted) average of the warped images. Assuming that the parameterization $\theta_n \rightarrow T^{\theta_n}$ is differentiable and that the transformation family is sufficiently well-behaved (as is the case, *e.g.*,



(a) A typical problem: if Ω_{scene} is not very large, the process is prone to a poor *global* minimum.



(b) A typical problem: drastic spatial changes in μ (note also that the end result is quite blurry).

Fig. 2: Typical problems in JA. Rightmost images are post-convergence results.

with the affine group or, more generally, spaces of diffeomorphisms [15,16,40,38,28]), the loss in Eq. (5) is differentiable. Thus, if θ_n is predicted using an STN (so, in particular, θ_n is a differentiable function of f^n , the STN’s input), the loss can, at least in principle, be minimized using standard DL training.

4 Identifying Key Challenges in Solving Joint-alignment Problems

Below we discuss three issues that might arise when solving JA problems: **1)** poor *global* minima; **2)** the need of regularization; **3)** the need of a good initialization.

Usually when trying to minimize a loss, reaching a global minimum is hard or even impossible, and if this feat happens to be achieved, it is deemed to be the ultimate success. Sadly, global minima of \mathcal{L}_{JA} , while being (very) easy to achieve, reflect, in fact, an ultimate failure; *e.g.*, the non-negative \mathcal{L}_{JA} can attain its global minimum (*i.e.*, zero) when all the frames are shrunk to an infinitesimally-small point. A similar phenomenon occurs if all the frames are warped outside Ω_{scene} (*e.g.*, see Figure 2a) or if the frames are warped such that there will be no pairwise overlap between them.

A popular solution in such cases is adding some type of regularization over $(\theta_n)_{n=1}^N$. However, while various forms of regularization have been suggested, each of them imposes a certain bias; *e.g.*, the regularization term in [31] favors symmetric distributions while the one in [10] pushes the (affine) transformations towards the Special Euclidean group, denoted by $\text{SE}(2)$. The implied assumptions in both these cases are limiting. Likewise, penalizing the size of the transformations (*e.g.*, by penalizing some norm of θ_n) is problematic when the accumulative motion of the camera is large, while regularization favoring temporal smoothness is not always compatible with real camera motions. Another issue is the need of hyperparameter tuning for the weight of the regularization term. Moreover, finding a combination of a regularization type and a weight that will work well for a sufficiently-large variety of videos is difficult.

JA is usually a difficult non-convex problem. Thus, a good initialization can be useful; *e.g.*, in JA-POLS [10] an STN-based JA module had to rely on an initialization based on SE-Sync [37]. The latter provides a useful globally-optimal solution to a different-but-related problem: the estimation of *absolute* transformations that are consistent as possible with noisy measurements of pairwise *relative* transformations

between pairs of frames, where both the latent absolute transformations and the observed relative ones are in $SE(2)$. With that initialization, the STN needs to solve an easier problem and does so over the more expressive Affine group. There are, however, several problems with the JA approach in [10] (we will later also discuss problems related to the background-modeling approach in [10]). First, pre-processing and heuristics are needed for extracting the relative transformations. Second, in cases where some of the true latent absolute transformations are far from $SE(2)$ (*e.g.*: when the video contains a significant accumulative variation in the distance between the camera and the scene; when the camera zoom is changing; when there is a strong perspective effect; *etc.*), the initialization breaks and this leads in turn to JA-POLS’ failure. Moreover, SE-Sync is neither robust nor differentiable w.r.t. the input frames. As there is no easy way to differentiate SE-Sync w.r.t. the input frames, the STN-based JA module in JA-POLS cannot be used in an end-to-end DL pipeline.

4.1 An Additional Challenge with Joint Alignment When Using Batches

Typically, due to the data size and as it is almost always the case in DL, the optimization is done batch by batch where each batch consists of a subset (selected at random) of the frames from the entire video. A single epoch then represents a full pass over the entire data, and the frames are reshuffled between epochs. This typically-necessary batch-by-batch processing creates an optimization difficulty which might appear to be minor but is, in fact, far more critical than it may seem (we will revisit this point in § 5.1). The issue is that the mean image μ (from Eq. (5)) is a function of the entire video, not just the frames in the current batch. A seemingly-obvious solution is to hold μ fixed during each epoch – so it does not affect the computation of the loss’ gradient – and then, at the end of each epoch, recompute μ . However, a problem that arises with that approach is that the difference between the alignment targets (that is, the previous μ and the recomputed one) in each pair of consecutive epochs might be large, making the optimization difficult since the optimal transformations for one target might be quite far from those that are optimal for the next target. For an illustration, see Figure 2b. A different approach, used in [10], picks the target μ to be the mean of only the (warped) frames in the current batch. Besides the fact that this is somewhat inconsistent with the cost-function formulation, that approach too can cause significant changes in the targets between consecutive batches. The jumping-target problem complicates the optimization more than one may expect. This is especially an issue at the beginning of the process when the frames are completely misaligned. For example, in retrospect, this is partly why JA-POLS [10] had to rely on the SE-Sync-based initialization scheme: as shown in [10], except in the simple case where the accumulative camera motion is small, without that initialization JA-POLS usually fails.

5 The Proposed Method: DeepMCBM

The proposed modules of joint alignment (using an STN) and background modeling (using a CAE) are presented in § 5.1 and § 5.2, respectively. Together, they form the proposed method, DeepMCBM. The goal of the STN straining is 1) to jointly align the

Algorithm 1: Training an STN for Joint Alignment

Input: $N_{\text{epochs}}, N_{\text{batches}}, \rho(\cdot), \text{data_loader}$
Data: $(f^n)_{n=1}^N$
Output: A trained STN for Joint Alignment

- 1 Initialize accumulators $\mathcal{G} \in \mathbb{R}^{H \times W \times C}$ and $\mathcal{M} \in \mathbb{R}^{H \times W}$ // see text
- 2 **for** $e \in \{1, \dots, N_{\text{epochs}}\}$ **do**
- 3 **for** $i \in \{1, \dots, N_{\text{batches}}\}$ **do**
- 4 $(f^b)_{b \in B} \leftarrow \text{data_loader}$ // Load batch: $B \subset \{1, \dots, N\}$
- 5 $(\theta_b, g^b)_{b \in B} \leftarrow \text{STN}((f^b)_{b \in B})$ // Note that $g^b = f^b \circ T^{\theta_b}$
- 6 $(M^b)_{b \in B} \leftarrow (M^\Omega \circ T^{\theta_b})_{b \in B}$ // Warp masks
- 7 $\mathcal{G}, \mathcal{M}, \mathcal{L}_{\text{batch}} \leftarrow \text{Algorithm 2}(\mathcal{G}, \mathcal{M}, (g^b)_{b \in B}, (M^b)_{b \in B})$ // Update \mathcal{G} and \mathcal{M} ; measure $\mathcal{L}_{\text{batch}}$ (*i.e.*, the batch loss)
- 8 Perform an optimization step to minimize the $\mathcal{L}_{\text{batch}}$ loss.
- 9 $(\mathcal{G}, \mathcal{M}) \leftarrow (\lambda \mathcal{G}, \lambda \mathcal{M})$ // Keep the history, but downweight it

video frames, implicitly forming a panoramic image, and 2) to learn how to warp an input frame towards that panoramic image. The goal of the CAE training is to learn the variability in the differences between the panoramic image and the input frames, while taking the warping into account but ignoring the foreground objects. The conditioning is done using the robust version of the panoramic pixelwise mean and variance.

5.1 A Regularization-free Strategy for Joint Alignment

Having identified, in § 4.1, that the jumps in the values of μ cause a major difficulty in the STN-based optimization of \mathcal{L}_{JA} (Eq. (5)), we design a simple but surprisingly-effective optimization strategy, summarized in Algorithm 1 (which, in turn, uses Algorithm 2 as its subroutine). During the training epochs, instead of computing μ using only the current batch (as was done in [10]), or instead of recomputing μ from scratch each epoch, we construct our μ from the warped frames in the current batch while also taking into account, albeit with a lower weight, all the warped frames from the previous epochs as well as the previous batches in the current epoch. The proposed algorithm uses accumulators, denoted by \mathcal{G} and \mathcal{M} . The former is used to accumulate weighted sums of the values of the pixels in the warped frames while the latter serves a similar purpose with the values of the pixels in the warped masks. Concretely, let e denote the index of the current epoch and let e' denote the index of some previous epoch. When evaluating the loss in a batch during epoch e , the contribution of the results from epoch e' becomes smaller and smaller as the “time” difference, $e - e'$, grows. This is done in line 9 in Algorithm 1 by multiplying the accumulators of the warped frames and the warped masks by a positive factor λ where $\lambda < 1$ (we use $\lambda = 0.9$).

As shown in Figure 3, the resulting targets (*i.e.*, the μ sequence formed during the optimization) change smoothly between epochs. Importantly, this behaviour has a profound and fourfold positive effect: **1. No complicated initialization is needed.** As the optimization becomes much easier, the initial transformations are simply taken to be the identity. **2. Regularization-free JA.** No form of regularization on $(\theta_n)_{n=1}^N$ is

Algorithm 2: Update $(\mu, \mathcal{G}, \mathcal{M})$ and measure the loss on the batch

Input: $\mathcal{G}, \mathcal{M}, (g^b)_{b \in B}, (M^b)_{b \in B}$
Output: $\mathcal{G}, \mathcal{M}, \mathcal{L}_{\text{batch}}$

- 1 $\mathcal{G} \leftarrow \mathcal{G} + \sum_{b=1}^B g^b$ // update warped-image accumulator
- 2 $\mathcal{M} \leftarrow \mathcal{M} + \sum_{b=1}^B M^b$ // update warped-mask accumulator
- 3 $\mu \leftarrow \mathbf{0}_{H \times W \times C}$
- 4 **for** $\mathbf{x} \in \{\mathbf{x} : \mathbf{x} \in \Omega_{\text{scene}} \text{ and } \mathcal{M}_{\mathbf{x}} \geq 0\}$ **do in parallel**
- 5 **for** $c \in \{1, \dots, C\}$ **do in parallel**
- 6 $\mu_{\mathbf{x},c} \leftarrow \frac{\mathcal{G}_{\mathbf{x},c}}{\mathcal{M}_{\mathbf{x}}}$
- 7 $\mathcal{L}_{\text{batch}} \leftarrow \frac{1}{B} \sum_{b=1}^B \frac{1}{C} \sum_{c=1}^C \left[\left(\sum_{\mathbf{x} \in \Omega_{\text{scene}}} M_{\mathbf{x}}^b \rho(g_{\mathbf{x},c}^b - \mu_{\mathbf{x},c}) \right) / \left(\sum_{\mathbf{x} \in \Omega_{\text{scene}}} M_{\mathbf{x}}^b \right) \right]$



(a) Compared with Figure 2a, the process is more stable and successful. Also, even when μ nears the border of Ω_{scene} , it never goes outside it.



(b) Compared with Figure 2b, the drastic jumps are eliminated. Also, with the proposed term the results are less affected by the specified size of Ω_{scene} .

Fig. 3: Results analogous to those in Figure 2 except they were obtained with the proposed memory-based approach. Rightmost images are post-convergence results.

needed; *e.g.*, there is no need to worry about the poor global minima from § 4. Since the optimization is gradient-based and since each epoch lingers in the “history” of the process for many epochs before its effective weight decays to zero (due to the repeating multiplications by $\lambda \in (0, 1)$), such undesired cases are eliminated altogether. For instance, as the stack of the original frames overlaid over each other (from the first epoch) contributes to the computation of μ , either shrinking the frames to a point or moving them outside Ω_{scene} will incur a loss. Our regularization-free JA is in sharp contrast to many algorithms including classical works (*e.g.*, [31]) and more recent ones (*e.g.*, [10]).

3. Higher expressiveness. The formulation lets us increase the expressiveness of the transformation family as needed. For example, JA-POLS is so crucially dependent on its SE-Sync initialization and SE-based regularization, that the affine transformations it predicts are nearly in SE(2) themselves. In contrast, our method can not only predict more general transformations in the Affine group but also use broader transformation families. In our experiments we demonstrate this using the group of homographies but one may also try richer STNs such as those based on diffeomorphisms [40, 1, 14].

4. Our JA module can be used in end-to-end pipelines. This is unlike not only non-DL methods but also JA-POLS [10] whose non-differentiable initialization prevents its JA

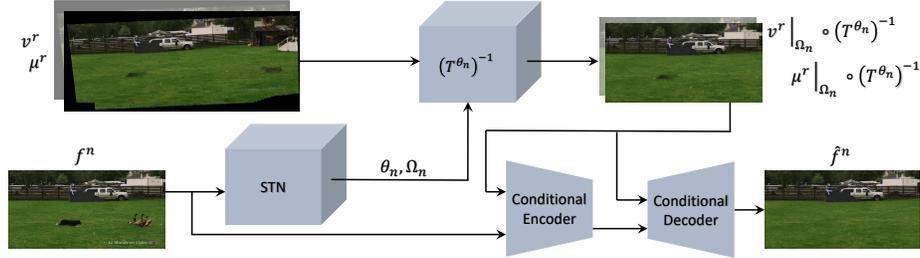


Fig. 4: The background-modeling module. After the STN module was trained using [Algorithm 1](#), the robust panoramic moments, μ^r and v^r , are computed. A CAE is trained for a robust reconstruction task, using the transformation parameters, $(\theta_n)_{n=1}^N$, estimated by the (frozen) STN. The CAE’s output is \hat{f}^n , the estimated background associated with f^n and the conditioning is done by (un)warping μ^r and v^r towards each input training image, f^n . During test time the process is similar, except that the transformation being used is the one predicted by the STN.

module from being used in an end-to-end manner. The technical details of the training process appear in our **Supplemental Material (Supmat)**.

5.2 Background Modeling in Ω (not Ω_{scene}) via a Conditional Autoencoder

Upon the training of the STN, the frames become jointly aligned. In principle, at this point all that is left to do is to learn a background model using either non-DL methods (*e.g.*, based on either pixelwise mixture models or RPCA methods; see [§ 2](#)) or deep ones (such as using a robust loss when training an autoencoder for reconstruction). However, there are several problems with this approach. First, it does not scale well: if the accumulating motion of the camera throughout the video is large, the panoramic image (of the entire scene covered throughout the video) can be huge. Moreover, in such a case even scalable RPCA methods will have to face an additional problem: since the domain of each warped image captures only a small region inside the domain of the panoramic image, it means that *most* of the pixels will represent missing data. Thus, one would need an RPCA method which can not only scale well but also succeed in situations where more than, say, 90%-95% of the data is missing. Also important is the following. Recall that given an input image, our goal is to estimate a background image, of the same size, that corresponds to that image. Thus, why should we even bother with trying to learn a panoramic-size background model? In [\[10\]](#), the discussion above motivated the learning of multiple local RPCA models and then, for estimating the background of a given image, only a subset of those models whose domains overlapped with the frame of interest were used. That solution, however, means that the number of models to be learned grows with the size of the panoramic image. Moreover, its non-DL formulation was another reason why JA-POLS was not an end-to-end method.

Here we propose a better alternative, whose pipeline is summarized in [Figure 4](#): use a CAE to learn a background model whose domain is small. This has two advantages:

1) It does not compromise the end-to-end nature of the method. 2) We need to learn only a single model (unlike in [10]) and its domain is small, fixed, and does not grow with the size of panoramic image (unlike in PRPCA). Concretely, rather than learning a background model (or models) whose domain is Ω_{scene} , we train a CAE on the original (*i.e.*, non-warped) input frames, using a robust reconstruction error and, for each input frame f^n , conditioning both the encoder and the decoder on (robust versions of) the mean and variance of the pixel stacks, but not before unwarping those central moments from Ω_{scene} towards f^n . We now provide the details. The first and second central moments, denoted by μ^r and v^r , respectively, are C -channel $H \times W$ images defined on Ω_{scene} and computed robustly using trimmed averaging as follows. Fix $\mathbf{x} \in \Omega_{\text{scene}}$, let $N_{\mathbf{x}} = |\{n : M_{\mathbf{x}}^n > 0\}|$, and let $(g_{\mathbf{x},c}^{(1)}, \dots, g_{\mathbf{x},c}^{(N_{\mathbf{x}})})$ be the order statistics of $p_{\mathbf{x},c}$. The values of μ^r and v^r at \mathbf{x} in channel c are computed, respectively, by

$$\mu_{\mathbf{x},c}^r = \frac{1}{(1-2\alpha)N_{\mathbf{x}}} \sum_{i=\alpha N_{\mathbf{x}}}^{(1-\alpha)N_{\mathbf{x}}} g_{\mathbf{x},c}^{(i)} \quad v_{\mathbf{x},c}^r = \frac{1}{(1-2\alpha)N_{\mathbf{x}}} \sum_{i=\alpha N_{\mathbf{x}}}^{(1-\alpha)N_{\mathbf{x}}} (g_{\mathbf{x},c}^{(i)} - \mu_{\mathbf{x},c}^r)^2. \quad (6)$$

Such trimmed averaging is a standard technique for computing robust moments [21]. The trimming parameter, α , was empirically set to $\alpha = 0.3$ as it provided a good balance between sample size and robustness. That said, the results when using any other value in the wide range between 20% and almost 50% were similar. Next, when f^n is fed into the CAE, the encoder and the decoder are conditioned by

$$\mu^n \triangleq (\mu^r|_{\Omega_n}) \circ (T^{\theta_n})^{-1} \quad \text{and} \quad v^n \triangleq (v^r|_{\Omega_n}) \circ (T^{\theta_n})^{-1} \quad (7)$$

which are $h \times w$ images (with C channels) defined on Ω and are nothing more than the portion of μ^r and v^r that is relevant for f^n . Using a code whose length was only 4, the CAE was trained with the following loss:

$$\mathcal{L}_{AE} = \sum_{n=1}^N \sum_{c=1}^C \sum_{\mathbf{x}' \in \Omega} \rho_{\text{recon}}(f_{\mathbf{x}',c}^n - \hat{f}_{\mathbf{x}',c}^n) \quad (8)$$

$$\hat{f}^n = \text{Decoder}(\text{Encoder}(f^n; \mu^n, v^n); \mu^n, v^n) \quad (9)$$

where \hat{f}^n is the output of the CAE and ρ_{recon} is a differentiable robust error function. We remark that, by design, the fact that μ^n and v^n are of the same dimensions as the input, f^n , also means it is easy to implement the conditioning via a convolutional layer. For more details about the CAE (whose architecture is based on the AE from [2]) as well as other training details, see our **Supmat**. Finally, ρ_{recon} should usually be more robust than ρ_{JA} . The reason is that while in JA the influence of foreground objects is relatively small, in the CAE-based reconstruction it is important, in every pixel, to eliminate the outliers (*i.e.*, the foreground pixels) as much as possible. Thus, we use the smoothed ℓ_1 loss (which is closely-related to Huber's function [5]) for ρ_{JA} and the Geman-McClure error function [17] for ρ_{recon} . See **Supmat** for details.

6 Results

We experimented with 4 variants of the proposed DeepMCBM: 1. **Basic/Aff**: This version uses only the STN-based JA module, without the CAE. It estimates the background

by simply unwarping the robust mean towards the input image. The transformations used in the STN belong to the Affine group (the invertibility of the transformations was guaranteed via the matrix exponential; see **Supmat**). 2. **CAE/Aff**: This version too uses the Affine STN but also uses the CAE module (for estimating the background). 3. **Basic/Hom** and 4. **CAE/Hom**: Similar to Basic/Aff and CAE/Aff, respectively, except that homographies are used instead of affine transformations. We compared those 4 variants with several methods: PRPCA [34]; JA-POLS [10]; PanGAEA [18]; DECOLOR [49]; PCP-PTI [9]; PRAC [19]. The 13 videos that we tested on are ones typically used for evaluation of methods in this area and are taken from well-known datasets [45,36]. Those movies cover camera motions in a variety of types, sizes, speed, zoom changes, *etc*. It should be noted that, due to their scalability limitations, PRPCA and PanGAEA could not run on the **ContinuousPan** video as the covered scene in the latter was too large. JA-POLS failed running on **zoomInZoomOut** (the significant zoom changes broke its key assumption). **Figure 5** contains a visual comparison, on select example videos, of DeepMCBM (in its CAE/Hom variant), PRPCA, JA-POLS, and PanGAEA. Results of the other (and less successful) methods (DECOLOR; PCP-PTI; PRAC), as well as more visual results (including videos) are in the **Supmat**.

Given an estimate of the background, subtracting it from the original frame yields a difference that can serve to determine foreground/background separation. To quantify the results in a threshold-independent way, for each method and each video we computed the Receiver Operating Characteristic (ROC) curve (using the ground truth) and its Area Under the Curve (AUC). The ROC curves are included in **Supmat**. We emphasize that our method is unsupervised and the ground truth information was used only for evaluation. **Table 1**, summarizing the AUC results, shows that DeepMCBM, especially with its CAE variants, is, overall, the leading method. In cases where DeepMCBM is not the first it is typically the runner-up. Moreover, unlike some competitors, DeepMCBM was applicable in all cases considered. The visual examples also illustrate how the CAE helps achieving a better estimate of the background than that one obtained by merely using the unwarped μ^r . We remark that our fixed code size, 4, is so small since: 1) the goal is not a typical reconstruction but to filter out foreground objects; 2) our AE is conditional so it is unsurprising a small size suffices. We could have made the code size video-dependent and thus improve results even further, but felt that a fixed size is simpler and makes a comparison with other methods fairer.

Predicting background for previously-unseen misaligned frames. In the comparison above, we focused on background/foreground estimation in the input videos on which the competing models (ours included) were learned. However, like JA-POLS, but unlike all the other methods, our method can predict the background in frames that were not included in the learning (more accurately, some of the competing methods can predict the background in the next constitutive frame, but they are unable to do so for misaligned frames in general such as those that are not consecutive). Due to space limits, we demonstrate that capability of DeepMCBM in the **Supmat**.

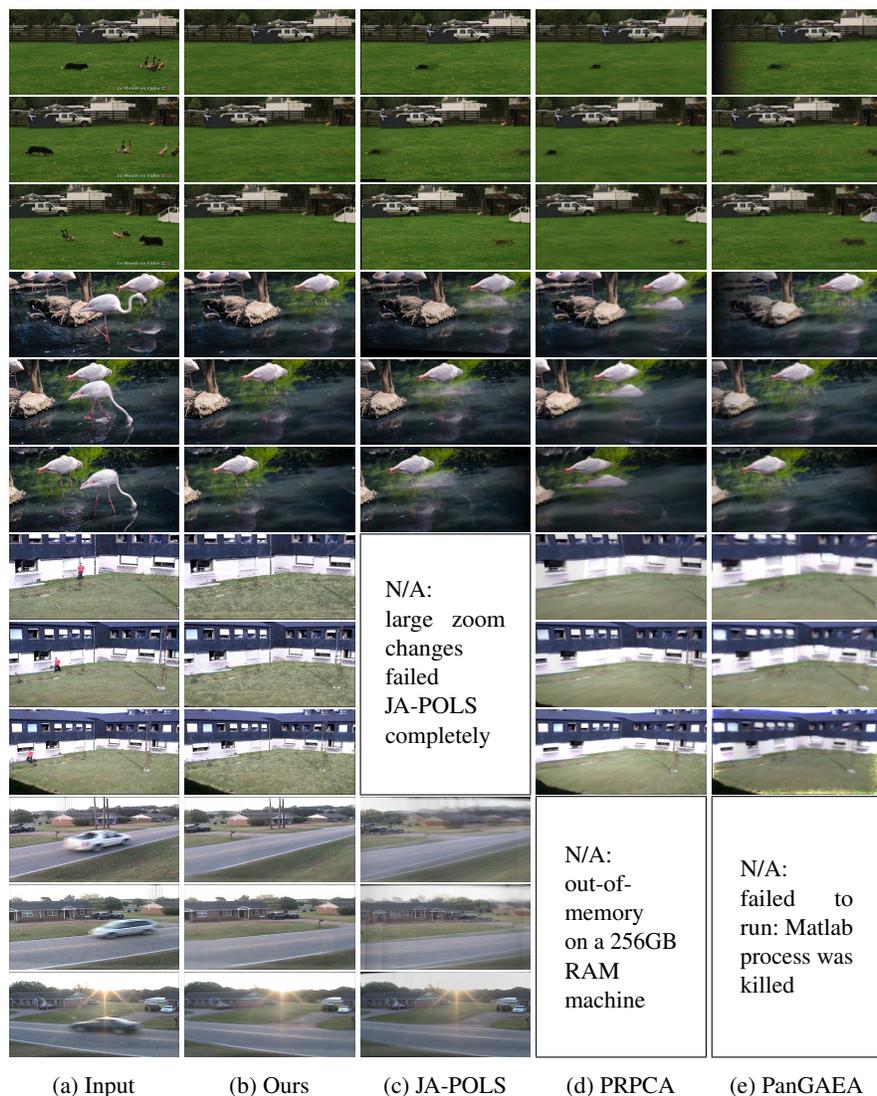


Fig. 5: Visual Comparison: Select Results. Note the ghosting artifacts and/or distortion in the other methods’ results. **Please zoom in to better appreciate the results.**

Ablation Study. As [Table 1](#) shows, the AE usually improves performance. In particular, its role is especially important when a foreground object spends a long portion of time in a static position (*e.g.*, the dog in the dog-geoses or the flamingo). In such cases, the robust mean alone still tends to capture some “ghosting” artifacts (as usually do all the competing methods) while the CAE helps correctly identifying that object as belonging to the foreground. The importance of the memory-based approach was

Sequence	DeepMCBM (Ours)				[34]	[18]	[9]	[10]	[19]	[49]
	Basic/Aff	CAE/Aff	Basic/Hom	CAE/Hom						
bmx-trees	.898	.896	.916	.908	.894	.786	.837	.930	.664	.737
boxing-fisheye	.924	.893	.927	.898	.935	.932	.728	.892	.627	.763
breakdance-flare	.931	.933	.953	.963	.960	.972	.740	.897	.806	.667
continuousPan	.897	.940	.895	.938	N/A	N/A	.846	.449	.656	.760
dog-geoses	.954	.984	.955	.984	.942	.917	.721	.947	.747	.886
flamingo	.962	.980	.961	.980	.891	.957	.638	.947	.560	.656
horsejump-high	.932	.942	.932	.943	.958	.908	.783	.914	.713	.892
sidewalk	.886	.908	.889	.932	.812	.702	.635	.851	.780	.935
stroller	.877	.885	.740	.756	.762	.904	.594	.807	.613	.721
stunt	.963	.979	.961	.978	.959	.954	.899	.930	.711	.781
swing	.880	.877	.887	.897	.942	.879	.805	.874	.722	.812
tennis	.960	.961	.959	.963	.943	.929	.831	.932	.787	.852
zoomInZoomOut	.981	.994	.981	.994	.979	.958	.720	N/A	.885	.957

Table 1: AUC scores for each method on each sequence.

also demonstrated in Figure 2 and Figure 3. In particular, the JA failures in Figure 2 imply that no subsequent background model could be built there, making a quantitative comparison (between using the memory term and not using it) a moot point. Finally, note that a basic (*i.e.*, unconditional AE) that knows nothing about the alignment has no chance here as it can only either simply reconstruct the entire frames (*i.e.*, with the undesired foreground objects) or fail in the reconstruction. Thus, when simply dropping the conditioning from our CAE, the resulting AE fails badly in background modeling; *e.g.*, its AUC for the Tennis video is 0.701 while DeepMCBM’s AUC score is 0.963.

7 Conclusion

The proposed DeepMCBM is an end-to-end DL solution for modeling background in a video from a moving camera. It supports a wide range of camera-motion types and sizes, scales gracefully, and achieves SOTA results. While we experimented with either affine transformations or homographies, DeepMCBM also supports more expressive transformations. The proposed regularization-free STN-based JA strategy may find usage in other applications, thereby the potential impact of this work may be broader than MCBMs. One limitation of our work is that, since DL involved, the training is slower in comparison to some competitors (JA-POLS excluded). However, we believe the SOTA results together with the other benefits DeepMCBM brings (end-to-end; scalability; the ability to predict background for previously-unseen misaligned frames; *etc.*) justifies it. The main failure case of the method is when foreground objects are large and much closer to the camera than the background is.

Acknowledgements. This work was supported in part by the Lynn and William Frankel Center at BGU CS and by Israel Science Foundation Personal Grant #360/21. G.E. was also funded by the VATAT National excellence scholarship for female Masters students in Hi-Tech-related fields.

References

1. Balakrishnan, G., Zhao, A., Sabuncu, M.R., Guttag, J., Dalca, A.V.: An unsupervised learning model for deformable medical image registration. In: CVPR (2018) [9](#)
2. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. In: ICLR (2017) [11](#)
3. Balzano, L., Nowak, R., Recht, B.: Online identification and tracking of subspaces from highly incomplete information. In: Allerton (2010) [4](#)
4. Berger, M., Seversky, L.M.: Subspace tracking under dynamic dimensionality for online background subtraction. In: CVPR (2014) [4](#)
5. Black, M.J., Rangarajan, A.: On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. IJCV (1996) [5](#), [11](#)
6. Brown, M., Lowe, D.G.: Automatic panoramic image stitching using invariant features. IJCV (2007) [4](#)
7. Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? JACM (2011) [3](#)
8. Chakraborty, R., Hauberg, S., Vemuri, B.C.: Intrinsic grassmann averages for online linear and robust subspace learning. In: CVPR (2017) [4](#)
9. Chau, G., Rodríguez, P.: Panning and jitter invariant incremental principal component pursuit for video background modeling. In: ICCV (2017) [4](#), [12](#), [14](#)
10. Chelly, I., Winter, V., Litvak, D., Rosen, D., Freifeld, O.: JA-POLS: a moving-camera background model via joint alignment and partially-overlapping local subspaces. In: CVPR (2020) [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [14](#)
11. Cox, M., Sridharan, S., Lucey, S., Cohn, J.: Least squares congealing for unsupervised alignment of images. In: CVPR (2008) [5](#)
12. Cox, M., Sridharan, S., Lucey, S., Cohn, J.: Least-squares congealing for large numbers of images. In: ICCV (2009) [5](#)
13. Cuevas, C., Moledano, R., García, N.: Statistical moving object detection for mobile devices with camera. In: ICCE (2015) [4](#)
14. Dalca, A., Rakic, M., Guttag, J., Sabuncu, M.: Learning conditional deformable templates with convolutional networks. In: NeurIPS (2019) [9](#)
15. Freifeld, O., Hauberg, S., Batmanghelich, K., Fisher III, J.W.: Highly-expressive spaces of well-behaved transformations: Keeping it simple. In: ICCV (2015) [6](#)
16. Freifeld, O., Hauberg, S., Batmanghelich, K., Fisher III, J.W.: Transformations based on continuous piecewise-affine velocity fields. IEEE TPAMI (2017) [6](#)
17. Geman, S., McClure, D.E.: Statistical methods for tomographic image reconstruction. In: BISI (1987) [11](#)
18. Gilman, K., Balzano, L.: Panoramic video separation with online Grassmannian robust subspace estimation. In: ICCV Workshops (2019) [4](#), [12](#), [14](#)
19. Guo, H., Qiu, C., Vaswani, N.: Practical reprocs for separating sparse and low-dimensional signal sequences from their sumpart I. In: ICASSP (2014) [4](#), [12](#), [14](#)
20. Guyon, C., Bouwmans, T., Zahzah, E.H.: Foreground detection via robust low rank matrix decomposition including spatio-temporal constraint. In: ACCV (2012) [3](#)
21. Hauberg, S., Feragen, A., Black, M.J.: Grassmann averages for scalable robust pca. In: CVPR (2014) [4](#), [11](#)
22. He, J., Balzano, L., Szlam, A.: Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. In: CVPR (2012) [4](#)
23. He, J., Zhang, D., Balzano, L., Tao, T.: Iterative grassmannian optimization for robust image alignment. Image and Vision Computing (2014) [4](#)

24. Huang, G., Mattar, M., Lee, H., Learned-Miller, E.G.: Learning to align from scratch. In: NIPS (2012) [5](#)
25. Huang, G.B., Jain, V., Learned-Miller, E.: Unsupervised joint alignment of complex images. In: ICCV (2007) [5](#)
26. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: NeurIPS (2015) [3](#)
27. Jin, Y., Tao, L., Di, H., Rao, N.I., Xu, G.: Background modeling from a free-moving camera by multi-layer homography algorithm. In: ICIP (2008) [4](#)
28. Kaufman, I., Weber, R.S., Freifeld, O.: Cyclic diffeomorphic transformer nets for contour alignment. In: ICIP (2021) [6](#)
29. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: A convolutional network for real-time 6-dof camera relocation. In: ICCV (2015) [4](#)
30. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: International Symposium on Mixed and Augmented Reality (2007) [4](#)
31. Learned-Miller, E.G.: Data driven image models through continuous joint alignment. IEEE TPAMI (2006) [5](#), [6](#), [9](#)
32. Meneghetti, G., Danelljan, M., Felsberg, M., Nordberg, K.: Image alignment for panorama stitching in sparsely structured environments. In: Scandinavian Conference on Image Analysis (2015) [4](#)
33. Miller, E.G., Matsakis, N.E., Viola, P.A.: Learning from one example through shared densities on transforms. In: CVPR (2000) [5](#)
34. Moore, B.E., Gao, C., Nadakuditi, R.R.: Panoramic robust PCA for foreground-background separation on noisy, free-motion camera video. IEEE Transactions on Computational Imaging (2019) [4](#), [12](#), [14](#)
35. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: Dtm: Dense tracking and mapping in real-time. In: ICCV (2011) [4](#)
36. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675 (2017) [12](#)
37. Rosen, D.M., Carlone, L., Bandeira, A.S., Leonard, J.J.: SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group. The International Journal of Robotics Research (2019) [6](#)
38. Shapira Weber, R., Eyal, M., Skaftte Detlefsen, N., Shriki, O., Freifeld, O.: Diffeomorphic temporal alignment nets. In: NeurIPS (2019) [6](#)
39. Sheikh, Y., Javed, O., Kanade, T.: Background subtraction for freely moving cameras. In: ICCV (2009) [4](#)
40. Skaftte Detlefsen, N., Freifeld, O., Hauberg, S.: Deep diffeomorphic transformer networks. In: CVPR (2018) [6](#), [9](#)
41. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: CVPR (1999) [3](#)
42. Szeliski, R.: Computer vision: algorithms and applications. Springer Science & Business Media (2010) [5](#)
43. Thurnhofer-Hemsi, K., López-Rubio, E., Domínguez, E., Luque-Baena, R.M., Molina-Cabello, M.A.: Panoramic background modeling for ptz cameras with competitive learning neural networks. In: IJCNN (2017) [4](#)
44. De la Torre, F., Black, M.J.: Robust principal component analysis for computer vision. In: ICCV (2001) [3](#)
45. Wang, Y., Jodoin, P.M., Porikli, F., Konrad, J., Benezeth, Y., Ishwar, P.: C3net 2014: an expanded change detection benchmark dataset. In: CVPR Workshop (2014) [12](#)
46. Wu, C.: Towards linear-time incremental structure from motion. In: International Conference on 3D Vision (2013) [4](#)

47. Xue, K., Liu, Y., Chen, J., Li, Q.: Panoramic background model for PTZ camera. In: International Congress on Image and Signal Processing (2010) [4](#)
48. Yalcin, H., Hebert, M., Collins, R., Black, M.J.: A flow-based approach to vehicle detection and background mosaicking in airborne video. In: CVPR (2005) [4](#)
49. Zhou, X., Yang, C., Yu, W.: Moving object detection by detecting contiguous outliers in the low-rank representation. TPAMI (2012) [4](#), [12](#), [14](#)
50. Zhou, Z., Li, X., Wright, J., Candes, E., Ma, Y.: Stable principal component pursuit. In: ISIT (2010) [3](#)