Spatiotemporal Bundle Adjustment for Dynamic 3D Human Reconstruction in the Wild

Minh Vo, Yaser Sheikh, and Srinivasa G. Narasimhan

Abstract—Bundle adjustment jointly optimizes camera intrinsics and extrinsics and 3D point triangulation to reconstruct a static scene. The triangulation constraint, however, is invalid for moving points captured in multiple unsynchronized videos and bundle adjustment is not designed to estimate the temporal alignment between cameras. We present a spatiotemporal bundle adjustment framework that jointly optimizes four coupled sub-problems: estimating camera intrinsics and extrinsics, triangulating 3D static points, as well as subframe temporal alignment between cameras and estimating 3D trajectories of dynamic points. Key to our joint optimization is the careful integration of physics-based motion priors within the reconstruction pipeline, validated on a large motion capture corpus of human subjects. We devise an incremental reconstruction and alignment algorithm to strictly enforce the motion prior during the spatiotemporal bundle adjustment. This algorithm is further made more efficient by a divide and conquer scheme with little loss in accuracy. We apply this algorithm to reconstruct 3D motion trajectories of human bodies in a dynamic event captured by uncalibrated and unsynchronized video streams in the wild. To make the reconstruction visually more interpretable, we fit a statistical human body model to the video streams. This fitting is constrained by the same motion prior, the 3D trajectory of the dynamic points, and other semantic cues extracted from the images. Because the videos are aligned with sub-frame precision, we reconstruct 3D motion at much higher temporal resolution than the input videos.

Index Terms—Spatiotemporal bundle adjustment, motion prior, temporal alignment, dynamic 3D reconstruction, human model fitting.

1 INTRODUCTION

When a moving point is observed from multiple cameras with simultaneously triggered shutters, the dynamic 3D reconstruction problem reduces exactly to the case of static 3D reconstruction. The classic point triangulation constraint [21], and the algorithmic edifice of bundle adjustment [34] built upon it, applies directly. Currently, there exists no consumer mechanism to ensure that multiple personal cameras, i.e., smartphones, consumer camcorders, or egocentric cameras, are simultaneously triggered [20]. Thus, in the vast majority of dynamic scenes captured by multiple independent video cameras, no two cameras see the 3D point at the same time instant. This fact trivially invalidates the triangulation constraint.

To optimally solve the dynamic 3D reconstruction problem, we must first recognize all the constituent sub-problems that exist. The classic problems of point triangulation and camera resectioning in the static case are subsumed. In addition, two new problems arise: reconstructing 3D trajectories of moving points and estimating the temporal location of each camera. Second, we must recognize that the sub-problems are tightly coupled. As an example, consider the problem of estimating 3D camera pose. While segmenting out stationary points and using them to estimate camera pose is a strategy that has been used in prior work [24], it ignores evidence from moving points that are often closer to the cameras and therefore provide tighter constraints for precise camera calibration. Imprecise camera calibration and quantization errors in estimating discrete temporal offsets result in significant errors in the reconstruction of moving points [14], [26], [38].

Prior work in dynamic 3D reconstruction has addressed some subset of these problems. For instance, assuming known (or separately estimated) camera pose and temporal alignment, Avidan and Shashua posed the problem of trajectory triangulation [4], where multiple noncoincidental projections of a point are reconstructed. Trajectory triangulation is an ill-posed problem and current algorithms appeal to motion priors to constrain reconstruction: linear and conical motion [4]; smooth motion [24], [36]; sparsity priors [45]; low rank spatiotemporal priors [29]. Estimating the relative temporal offsets of videos captured by the moving cameras is more involved [15], [42]. Currently, the most stable temporal alignment methods require corresponding 2D trajectories as input [9], [10], [23], [33], [40] and rely purely on geometric cues to align the interpolated points along the trajectories across cameras. Recent work has considered the aggregate problem, but address the spatial and temporal aspects of the problem independently [6], [17], [44].

In this paper, we introduce the novel concept of spatiotemporal bundle adjustment that jointly optimizes for all sub-problems simultaneously. Just as with static 3D reconstruction, where the most accurate results are obtained by jointly optimizing for camera parameters and triangulating static points, the most accurate results for dynamic 3D reconstruction are obtained when jointly optimizing for the spatiotemporal camera parameters and triangulating both static and dynamic 3D points. Unlike traditional bundle adjustment, we recognize the need for a motion prior in addition to the standard reprojection cost that jointly estimates the 3D trajectories corresponding to the sub-frame camera temporal alignment.

Consider this example: when a person jogging at 10m/s is captured by two cameras at 30Hz, one static and one handheld jittering at 3mm per frame, with the camera baseline of 1m, recording from 4m away. A simple calculation suggests that a naïve attempt to triangulate points of the static camera with their correspondences of the best-aligned frame in the other camera results in up to 40 cm reconstruction error.
We evaluate several physics-based 3D motion priors (least kinetic energy, least force, and least action) on the CMU motion capture repository [1]. Such joint estimation is most helpful for dynamic scenes with large background/foreground separation where the spatial calibration parameters estimated using background static points are unavoidably less accurate for foreground points.

Direct optimization of the spatiotemporal objective is hard and is susceptible to local minima. We address this optimization problem using an incremental reconstruction and temporal alignment algorithm. This optimization framework ensures the proposed 3D motion prior constraint is satisfied. Our algorithm naturally handles the case of missing data (e.g., when a point is occluded in a particular time instant) and scales to many cameras. Thus, we can produce accurate 3D trajectory estimation at much high temporal resolution than the frame rates of the input videos.

While the incremental reconstruction and alignment approach is effective and accurately optimizes the spatiotemporal bundle adjustment problem, it is not efficient. The computational complexity grows quadratically with the number of cameras. We solve this issue by dividing the optimization problem into overlapping groups of cameras with overlapping field of view, each of which is optimized independently using the incremental reconstruction and alignment scheme. These sub-problems are merged and globally optimized in the final pass. Empirically, this approach is at least 20 times faster and has marginal accuracy loss on our datasets.

We apply spatiotemporal bundle adjustment to reconstruct human dynamic scenes. While this algorithm can accurately reconstruct the 3D trajectory of the dynamic points, those points are usually sparse and hence, can be hard to visually interpret. We fit a statistical 3D human body model [22] to the unsynchronized and low frame rate videos to augment the reconstruction. This is in a similar spirit of Multiview stereo to sparse bundle adjustment [13], [28]. Due to the lack of triangulation constraints used in previous work [18], we employ the same physics-based motion prior and the sparse dynamic points to further constrain the fitting. Additionally, since we fit the mesh model to multiple unsynchronized videos simultaneously, unless the frame sequencing is properly estimated, the fitted mesh motion will contain significant jitters and loops (see supplementary video). This is the key difference between unsynchronized multiple cameras model fitting and monocular [16], [19], [35] or image-based model fitting [7], [25]. While ideally we should re-optimize camera calibration parameters and 3D points jointly with the body shape and pose coefficients, the extracted semantic cues are often imprecise and hurt the spatiotemporal bundle adjustment. Thus, we fix the estimated spatiotemporal parameters during the shape fitting. As a demonstration, we reconstruct 3D trajectories and human body shape of dynamic actions captured outdoor by ten smartphones without any constraints.

This paper extends our previous work [37] in multiple aspects. First, we provide a more thorough evaluation of the motion priors. Second, we devise a divide and conquer algorithm to speed up the incremental reconstruction and alignment framework. Third, we introduce a self-calibration method to estimate the rolling shutter readout speed and the spatial pose of a moving rolling shutter camera. Lastly, we build an end-to-end framework for human shape fitting to unsynchronized low framerate video cameras.

## 2 Motion Prior for Dynamic 3D Capture

Consider the scenario of $C$ video cameras observing $N$ 3D points over time. The relation between the 3D point $X^n(t)$ and its 2D projection $x^n_c(f)$ on camera $c$ at frame $f$ is given by:

$$
\begin{bmatrix}
x^n_c(f)
\end{bmatrix}
= K_c(f) \begin{bmatrix} R_c(f) & T_c(f) \end{bmatrix} \begin{bmatrix} X^n(t) \\
1 \end{bmatrix},
$$

(1)

where $K_c(f)$ is the intrinsic camera matrix, $R_c(f)$ and $T_c(f)$ are the relative camera rotation and translation, respectively. For simplicity, we denote this transformation as $x^n_c(f) = \pi_c(f, X^n(t))$. The time corresponding to row $r_c$ at frame $f$ is related to the continuous global time $t$ linearly: $f = \alpha_c t + \beta_c + \gamma_c \ast r_c$, where $\alpha_c$ and $\beta_c$ are the camera frame rate and time offset, $\gamma_c$ is the rolling shutter pixel readout speed. For global shutter camera, $\gamma_c$ is zero.

### Image reprojection cost

At any time instance, the reconstruction of a 3D point must satisfy Eq. 1. This gives the standard reprojection error $S_f$, which we accumulate over all 2D points observed by all $C$ cameras for all frames $F_c$:

$$
S_f = \sum_{c=1}^{C} \sum_{n=1}^{N} \sum_{f=1}^{F_c} I^n_c(f) \frac{\| \pi_c(f, X^n(t)) - x^n_c(f) \|^2}{\sigma^n_c(f)},
$$

(2)

where, $I^n_c(f)$ is a binary indicator of the point-camera visibility, and $\sigma^n_c(f)$ is a scalar, capturing the uncertainty in localizing $x^n_c(f)$ to $S_f$. Since the localization uncertainty of an image point $x^n_c(f)$ is proportional to its scale [43], we use the inverse of the feature scale as the weighting term for each residual term in $S_f$.

However, Eq. 2 is purely spatially defined and does not encode any temporal information about the dynamic scene. Any trajectory of a moving 3D point must pass through all the rays corresponding to the projection of that point in all views. Clearly, there are infinitely many such trajectories and each of these paths corresponds to a different temporal sequencing of the rays. Yet, the true trajectory must also correctly align all the cameras. This motivates us to investigate a motion prior that ideally estimates a trajectory that corresponds to the correct temporal alignment. The cost of violating such a prior $S_M$ can be then added to the image reprojection cost to obtain a spatiotemporal cost function that jointly estimates both the spatiotemporal camera calibration parameters and the 3D trajectories:

$$
S = \arg \min_{X(t), \{K, R, t\}, \alpha, \beta} S_f + S_M.
$$

Given multiple corresponding 2D trajectories of both the static and the dynamic 3D points $\{x_c(t)\}$ for $C$ cameras, we describe how to jointly optimize Eq. 3 for the 3D locations $X(t)$, the spatial camera parameters at each time instant $\{K_c(f), R_c(f), T_c(f)\}$ and the temporal alignment between cameras $\beta$. We assume the frame rate $\alpha$ is known.

### 2.1 Physics-based Motion Priors

In this section, we investigate several forms of motion prior needed to compute $S_M$ in Eq. 3. We validate each of these priors on the entire CMU Motion Capture Database [1] for their effectiveness on modeling human motion.

When an action is performed, its trajectories must follow the paths that minimize a physical cost function. This inspires the investigation of the following three types of priors: least kinetic energy, least force\(^2\), and least action [11]. See Fig. 1 for the formal definition of these priors. In each of these priors, $m$ denotes the

2. We actually use the square of the resulting forces.
mass of the 3D point, \( g \) is the gravitational acceleration force acting on the point at height \( h(t) \), and \( v(t) \) and \( a(t) \) are the instantaneous velocity and acceleration at time \( t \), respectively.

Mathematically, the least kinetic energy prior encourages constant velocity motion, the least force prior promotes constant acceleration motion, and the least action prior favors projectile motion. While none of these priors hold for an active system where forces are arbitrarily applied during its course of action, we conjecture that the cumulative forces applied by both mechanical and biological systems are sparse and over a small duration of time, the true trajectory can be approximated by the path that minimizes the costs defined by our motion priors. Any local errors in the 3D trajectory, either by inaccurate estimation of points along the trajectory or wrong temporal sequencing between points observed across different cameras, produce higher motion cost.

**Least kinetic motion prior cost:** We accumulate the cost over all \( N \) 3D trajectories for all time instances \( T^n \):

\[
S_M = \sum_{n=1}^{N} \sum_{i=1}^{T^n-1} w_n(t) \frac{m_n}{2} v_n(t)^2 (t^{i+1} - t^i),
\]

(4)

where \( \gamma_n(t) \) is the weighting scalar and \( m_n \) is the point mass, assumed to be identical for all 3D points and set to be 1. We approximate the instantaneous speed \( v(t) \) at time \( t^i \) along the sequence \( X^n(t) \) by a forward difference scheme, \( v_n(t^i) \approx \frac{X^n(t^{i+1}) - X^n(t^i)}{t^{i+1} - t^i} \). We add a small constant \( \epsilon \) to the denominator to avoid instability caused by 3D trajectories observed at approximately same time. Eq. 4 is rewritten as:

\[
S_M = \sum_{n=1}^{N} \sum_{i=0}^{T^n-1} w_n(t) \left\| \frac{X^n(t^{i+1}) - X^n(t^i)}{t^{i+1} - t^i + \epsilon} \right\|^2 (t^{i+1} - t^i),
\]

(5)

Using the uncertainty \( \sigma_n^2(f) \) of the 2D projection of 3D point \( X_n(t) \), the weighting \( w_n(t) \) can be approximated by a scaling factor that depends on the point depth \( \lambda \) and the scale \( \mu \), relating the focal length to the physical pixel size, as \( w_n = \frac{\lambda}{\sigma_n^2} \). The least force and least action prior costs can be computed similarly.

### 2.2 Evaluation on Motion Capture Data

Consider a continuous trajectory of a moving point in 3D. Sampling this continuous trajectory starting at two different times produces two discrete sequences in 3D. We first evaluate how the motion prior helps in estimating the temporal offset between the two discrete sequences. We extend this to 2D trajectories recorded by cameras later. The evaluation is conducted on the entire CMU marker-based Motion Capture Database [1] containing over 2500 sequences of common human activities such as playing, sitting, dancing, running and jumping, captured at 120 fps.

Each trajectory is subsampled starting at two different random times to produce the discrete sequences. 3D zero mean Gaussian noise is added to every point along the discrete trajectories. The ground truth time offsets are then estimated by a linear search and we record the solution with the smallest motion prior cost. For our test, the captured 3D trajectories are sampled at 12 fps and the offsets are varied from 0.1 to 0.9 frame interval in 0.1 increments.

As shown in Fig. 1, the least kinetic energy prior and least force prior perform similarly in this setting and both estimate the time offset between the two trajectories well for low noise levels. When more noise is added to the trajectory sequences, the sequencing is noisier. Yet, our motion cost favors correct camera sequencing over closer time offset. This is a desirable property because wrong sequencing results in a trajectory with loops (see Fig. 5). In contrast, the least action prior gives biased results even when no noise is added to the 3D data.

We further compare the sequencing expressiveness of the least kinetic energy prior to the least force prior for different cumulative frame rates. The cumulative frame rate is defined as the framerate of the virtual camera consists of all the frames from each camera. As shown in Fig. 2, the least force prior is more expressive than the least kinetic prior for lower frame rates. This is expected since the least force prior captures more local information of the trajectory. However, for modern video cameras, the cumulative framerate easily exceeds 120fps, where the alignment results using either priors are similar. Thus, we only use the least kinetic prior in the remainder of the paper. Extension to the least force is straight forward.

### 3 Spatiotemporal Bundle Adjustment

Unlike traditional bundle adjustment [34], the spatiotemporal bundle adjustment must jointly optimize for four coupled problems: camera intrinsics and extrinsics, 3D locations of static points, temporal alignment of cameras and 3D trajectories of dynamic points. However, direct optimization of Eq. 3 is hard because:

- (a) it requires solution to a combinatorial problem of correctly sequencing all the cameras
- (b) motion prior cost is strongly discontinuous as small changes in time offsets can switch the temporal ordering of cameras. Thus, it is not possible to ensure the satisfaction of the motion prior constraint.

We solve this problem using an incremental reconstruction and alignment approach where the camera is sequentially added to the optimization problem. This algorithm is further speed up by
a divide and conquer scheme where the groups of cameras are solved independently first and then merged and refined globally using continuous second order optimization. We initialize temporal alignment and the 3D trajectory of the dynamic points using a geometry (or triangulation constraint) based method [10], [30]. Even though the triangulation constraint is not strictly satisfied, empirically, the estimations provide a good starting point for the incremental reconstruction and alignment.

3.1 Incremental Reconstruction and Alignment

Our incremental reconstruction and alignment (IRA) approach adds camera one at a time. For every new camera, a linear search for the best sequencing of this camera with respect to the previous cameras based on the motion prior is conducted. Once the sequencing order is determined, we use continuous optimization to jointly estimate all the spatiotemporal camera parameters, and static points and dynamic trajectories. Thank to the linear search step, we can enforce the motion prior constraint strictly without any discontinuities due to incorrect time ordering of cameras. We summarize this method in Algorithm 1.

3.1.1 Temporal alignment of two cameras

We refine the initial guess by optimizing Eq. 3. However, just as in point triangulation, the 3D estimation from a stereo pair is unreliable. Thus, we simply do a linear search on a discretized set of temporal offsets and only solve Eq.3 for the 3D trajectories. The offset with the smallest cost is taken as the sub-frame alignment result. We apply this refinement to all pair of cameras.

3.1.2 Which camera to add next?

As in incremental SFM [12], [30], we need to determine the next camera to include in the calibration and reconstruction process. For this, we create a graph with each camera as a node and define the weighted edge cost between any two cameras $i^{th}$ and $j^{th}$ as

$$E_{ij} = \sum_{k=1, k \neq i,j}^{C} S_{ij} \left| t_{ij} + t_{jk} - t_{ik} \right| / N_{ij} B_{ij},$$

(6)

where $t_{ij}$, $N_{ij}$, $B_{ij}$, and $S_{ij}$ are the pairwise offset, the number of visible corresponding 3D points, the average camera baseline, and the spatiotemporal cost evaluated for those cameras, respectively. Intuitively, $|t_{ij} + t_{jk} - t_{ik}|$ encodes the constraint between the time offsets among a camera triplet, and $N_{ij} B_{ij}$ is a weighting factor favoring the camera pair with more common points and larger baseline.

Similar to [10], [39], a minimum spanning tree (MST) of the graph is used to find the alignment of all cameras. We use the Kruskal MST, which adds nodes with increasing cost at each step. The camera processing order is determined once from the connection step of the MST procedure.

3.1.3 Estimating the time offset of the next camera

We temporally order the currently processed cameras and insert the new camera into possible time slots between them, followed by a nonlinear optimization to jointly estimate all the offsets and 3D trajectories. Any trials where the relative ordering between cameras change after the optimization are discarded, ensuring that the motion prior is satisfied. The trial with the smallest cost is taken as the temporal alignment and 3D trajectories of the new set of cameras.

3.2 Divide and Conquer

While the incremental reconstruction and alignment approach offers a tractable solution for optimizing Eq. 3, its complexity increases quadratically with the number of cameras. For every new camera, we must optimize Eq. 3 $C - 1$ times, where $C$ is the number of camera being processed, to determine the sequencing order with the least motion cost. To address the computational efficiency issue, we propose a divide and conquer approach to speed up solver with minimal loss in reconstruction accuracy. This algorithm is based on the observation that the temporal alignment becomes stable after a small number of cameras is processed (4 cameras in all of our experiments).

This algorithm proceeds in three stages. First, we form the camera groups with large co-visibility with them by creating a skeleton graph using camera graph built in Sec. 3.1.2 [31]. Here, each group is taken as two camera nodes in the skeleton graph and their connected cameras in the graph of Sec. 3.1.2. We purposely let the overlapping groups share two cameras to better detect and discard temporal inconsistency when merging all the groups together. Second, we process each camera group independently processed using the incremental reconstruction and alignment approach. For every pair of inconsistency groups detected, these groups are merged and re-processed using the first approach. Third, we aggregate to temporal alignment parameters from all groups into a common timeline and optimize all cameras jointly for the spatiotemporal calibration parameters and the 3D position of the static and dynamic points.

3.3 Motion Resampling via Discrete Cosine Transform

Note that Eq. 5 approximates the speed of the 3D point using finite difference. While this approximation allows better handling of missing data, the resulting 3D trajectories are often noisy.
3.4 Evaluation on Motion Capture Data

We validate the proposed spatiotemporal bundle adjustment on synthetic data generated from the CMU Motion Capture database [1]. We sequentially distribute the ground truth trajectory, captured at 120 fps, to 10 global shutter perspective cameras with a resolution of 1920x1080 and 12fps. All cameras are uniformly arranged in a circle and capturing the scene from 3 m away. We randomly add 3000 background points arranged in a cylinder of radius 15 m centered at dynamic points. The relative offsets, discretized at 0.1 frames, are randomly varying for every sequence. None of the offsets generates cameras observing the 3D points synchronously. We assume that the initial offsets are within 2-3 frames accurate, which is the case for most geometry-based alignment methods. We also add zero mean Gaussian noise of 2 pixels standard deviation to the observed 2D trajectories. For the divide and conquer approach, we split the camera into 3 groups of 4 nearby cameras each.

The reconstruction and alignment errors are summarized in Fig. 3 and Table 1. Spatially, the point triangulation of the frame-accurate alignment propagates the error to all cameras and gives the worst result. Trajectories reconstructed using 3D cubic B-spline basis gives much smaller error than the point triangulation. However, it also arbitrarily smooths out the trajectories and is inferior to our method. While both the direct motion prior and DCT resampling have similar mean error (direct: 6.6 cm, DCT: 6.5 cm), the former has a larger maximum error due to the noise in approximating the velocity. Temporally, our method can estimate ground truth offset at sub-frame accuracy with low uncertainty. The divide and conquer approach is quantitatively equally accurate with the incremental reconstruction and alignment method while being approximately 30 times faster. We also observe that this approach produces no temporal inconsistency between camera groups for all trials.

4 DYNAMIC RECONSTRUCTION OF HUMAN BODY

While the spatiotemporal bundle adjustment can accurately estimate the 3D trajectory of dynamic points, the number of such trajectories is sparse and is insufficient to fully visualize the scene content. In this section, we leverage recent advances in semantic understanding to estimate the 3D human body shape and pose by fitting a statistical body model to the observed semantic cues. More specifically, the optimization is constrained by the 2D human body part segmentation from each image, the body anatomical keypoints, the sparse 3D trajectories recovered using sparse spatiotemporal bundle adjustment, the least kinetic energy motion prior, and the spatial pose and shape priors. Due to the imprecision in localizing the semantic cues, we fix camera parameters and only optimize for the 3D human body.

4.1 Statistical Body Model

We use the SMPL model [22], a linear blend shape model of body shape that can be deformed via linear blend skinning, to present the human body. This model $V(\Omega, \Phi, \Gamma)$ is a triangle mesh, composed of 6890 vertices, and is parameterized by gender, 10 identify shape coefficients $\Omega$, 24 joints $\Phi$, presented using angle-axis to model the relative rotation between body parts, and a body root translation $\Gamma$. The 3D location of the body joints $X$ corresponding to a particular pose configuration are given by the joint regressor $R$, a matrix presenting a sparse linear combination of surface vertices around the joint, $\tilde{X} = RV(\Omega, \Phi, \Gamma)$. In our case, $R$ is slightly different from [22] as our joints are defined according to the OpenPose keypoints format [8].

### Table 1: The reprojection error for the entire CMU Mocap dataset. MP is the results for all cameras without resampling (R).

<table>
<thead>
<tr>
<th>Geometric</th>
<th>Incremental</th>
<th>Divide and conquer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometry</td>
<td>MP</td>
<td>R</td>
</tr>
<tr>
<td>Static</td>
<td>3.45</td>
<td>2.93</td>
</tr>
<tr>
<td>Dynamic</td>
<td>17.8</td>
<td>1.68</td>
</tr>
<tr>
<td></td>
<td>0.85</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>0.74</td>
<td>0.89</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td></td>
</tr>
</tbody>
</table>
4.2 Body Alignment Objective

We fit the SMPL model to the observed semantic body part and keypoint detectors and the sparse 3D trajectory by optimizing the following cost:

\[
S = \arg \min_{\Theta, \Phi(t), \Gamma(t)} \lambda_{S_J} S_J + \lambda_{S_T} S_T + \lambda_{S_M} S_M + \lambda_{S_B} S_B,
\]

where \( \{S_J, S_T\} \) are the image evidence cost, capturing the body joint, and image silhouette, respectively, \( S_T \) is the cost induced by the sparse 3D trajectory on the body, \( S_M \) is the least kinetic motion prior loss imposed on the 3D body joints, and \( S_B \) is the body pose and shape prior cost. We normalize each of these costs by the number of their contributing residuals before applying the weights \( \lambda_{S_{K}}, \lambda_{S_T}, \lambda_{S_S}, \lambda_{S_M}, \lambda_{S_B} \) to each cost. The definition of these cost functions are described as follow.

**Body keypoint alignment cost:** This cost function aims to reduce the difference between the projected SMPL keypoints and the detected keypoints, and is written as

\[
S_J = \sum_{c=1}^{C} \sum_{f=1}^{F_c} \sum_{j=1}^{J} I_c^j(f) \sigma_j^2(f) \rho \left( \frac{||\pi_c(f, X^j) - x_c^j(f)||}{\sigma_j} \right),
\]

where \( I_c^j(f) \) is a scalar approximating the uncertainty in detecting the body keypoints, \( X_j \) is a joint among the set \( X(t) = \mathcal{R}(\Theta, \Phi(t), \Gamma(t)) \), \( I_c^j(f) \) is a binary indicator of the point visibility, and \( \sigma_j^2(f) \) is the confidence of the keypoints detected by OpenPose [8].

**Sparse 3D trajectory constraints:** This cost function penalizes variance of the distances \( L(., .) \) between the point \( X_n \) along the 3D trajectories to the two nearest joints within the same body part, \( X^j \in 2N(N(X^j)) \), and is expressed as

\[
S_T = \sum_{c=1}^{C} \sum_{f=1}^{F_c} \sum_{n=1}^{N} \sum_{j=1}^{J} I_c^j(f) \rho \left( \frac{||L(X^j, X_n) - \mathcal{L}(X^j, X_n)||^2}{\sigma_T} \right),
\]

where \( \sigma_T \) is a scalar capturing the uncertainty in estimating the location point along the sparse 3D trajectory, \( I_c^j(f) \) is a binary showing the availability of \( X_n(t) \) on to camera \( f \) at frame \( f \), and \( l \) is the Euclidean distance between two points. Here, we add the extra variable \( l(., .) \) as the average distance between points over the entire course of motion, to the optimization. Despite the non-rigid body and cloth deformation, we expect the deviation of the instantaneous point distance to be close to its average over time. Empirically, we observe that this loss function improves tracking robustness especially in case of semantic detector failure when body parts corresponding different person are grouped together.

**Silhouette alignment cost:** This cost function discourages any projected body vertices not contain inside the detected body segmentation and is expressed as

\[
S_S = \sum_{c=1}^{C} \sum_{f=1}^{F_c} \sum_{v \in V(\Theta, \Phi(t), \Gamma(t))} \rho \left( \frac{\text{DT}_c(f, \pi_c(f, v))}{\sigma_P} \right),
\]

where \( \text{DT}_c(f) \) is the distance transform of the body part segmentation in camera \( c \) at frame \( f \), \( \text{DT}_c(f) \) is zeros for points inside projected mesh and is equal to the distance between the projected mesh vertex and its nearest point on the segmentation boundary contour otherwise. This loss is particularly useful for occluded body parts. We use DensePose [3] to compute body segmentation.

**Motion prior cost:** Due to the lack of triangulation constraints in unsynchronized camera setup, the motion prior is key to for accurate dynamic human body estimation, especially for occluded body parts under fast body motion and observed by low frame rate cameras. Similar to sparse 3D trajectory estimation, we use the least kinetic motion prior with forward differentiation approximation of the velocity imposed on the 3D body joints \( X^j \) to constraint to body motion over its entire observation duration \( T \)

\[
S_M = \sum_{t \in T} \sum_{j} \left( \frac{||p(r + 1) - p(r)||^2}{\sigma_M} \right),
\]

where \( \sigma_M \) is the expected variation in human instantaneous velocity. We set \( \sigma_M \) differently for different joints.

**Body shape prior cost:** The SMPL model is designed to fit semi-naked people, whereas we are interested in measuring people who are wearing arbitrary clothes and accessories. Moreover, the semantic detector that the model is being fit to could produce erroneous estimation, especially for occluded body parts. Such limitation of the body model and the incorrect correspondences can significantly affect both the body shape and pose parameters. We mitigate such defects by placing a zero-mean standard normal distribution over the pose \( \Phi(f) \) (favors mean pose) and shape \( \Omega \) parameters as

\[
S_B = \lambda_{\Omega} N(\Omega) + \lambda_{\Phi} \sum_{c=1}^{C} \sum_{f=1}^{F_c} N(\Phi(f)),
\]

where \( \{\lambda_{\Omega}, \lambda_{\Phi}\} \) are weighting scalar between the residuals.

4.3 Optimization Strategy

Due to the complexity of the human body pose, a direct optimization of Eq. 9 converges slowly and often fails to produce accurate body fitting. We solve the problem in three stages. (1) full sequence spatiotemporally coherent 3D human skeleton estimation (2), per-time-instance human model fitting to the skeleton, and (3) window-based accurate and temporally coherent body pose and shape fitting. These stages are described below.

**Stage 1: Coherent 3D human skeleton estimation** For each person in the scene, we wish to estimate a temporally and physically consistent human skeleton model for the entire sequence. This is achieved by minimizing an energy function that combines of the reprojection cost on the detected semantic keypoint of Eq. 10, the least kinetic motion prior cost of Eq. 10, and the prior on human skeleton written as

\[
S_b = \sum_{t \in T} \sum_{q \in Q} \left( \frac{L(q, t) - L(q)}{\sigma_L} \right)^2,
\]

\[
S_{tr} = \sum_{t \in T} \sum_{(r, t) \in S} \left( \frac{L(r, t) - L(r, t)}{\sigma_S} \right)^2,
\]

where \( Q \) is the set of keypoint connectivity within all rigid body parts, \( S \) denotes the set of joints of the corresponding left and right limb, \( \{\sigma_L, \sigma_S\} \) captures the precision of the symmetry and left-right constancy constraints. These priors enforce the left-right symmetry of the body bone length and penalize large changes between the bone length estimated each time instance and average bone length \( \mathcal{L} \) over the entire sequence. As in Sec. 3, the initial 3D skeleton is obtained by geometric triangulation. We weight the
the framerate to simulate faster motion, which invalidates the
geometry constraints for unsynchronized cameras and stresses
the essence of the motion prior: 10 fps, 30 fps, and 15 fps for
Checkerboard, Jump, and Dance, respectively.

3D corpus and initial camera pose estimation: We track SIFT
features using affine optical flow [5] and sample keyframes,
defined as frames where the number of tracked features drop 40%
from the last key frame, from all videos. These keyframes are
passed to a SfM pipeline [27], [41] to build the 3D corpus of
the scene. We register other frames to this corpus using the
r6P algorithm and refine their parameters using the Cayley transform
model [2]. No temporal regularization is performed during the
registration to preserve the abrupt a camera motion frequently
observed due to the camera holder’s footstep.

Rolling shutter scanning speed estimation: Consider a moving
camera observing static features. Geometrically, this camera can
also be viewed as being static and observing moving features.
We estimate the camera rolling shutter readout speed by assuming
the 3D location of these moving features also obeys the
least kinetic motion prior for the duration of 1 frame. Denote
\( X_\nu(f), X_\nu(f+1) \) as the virtual location of the static feature
captured exactly at the first row of of frame \( f, f+1 \), respectively,
and \( X_\nu(t) \) is the true location of the same feature observed in the
image, \( t \in [t(f), t(f+1)] \). Under the least kinetic assumption
(e.g., constant velocity) and vertical rolling shutter readout, we
can present the 3D location of the observed feature as
\[
X(t) = X_\nu(f) + \gamma \frac{r}{h} (X_\nu(f+1) - X_\nu(f)),
\]
where \( r \) is the image row of the observed feature and \( h \) is the
image height. Using this representation of \( X \) to optimize Eq. 2,
we can estimate the rolling shutter readout speed \( \gamma \) for each camera.

Corresponding 2D trajectory generation: We detect and match
SIFT features across cameras at evenly distributed time instances.
We discard matches with low gradient score and track the re-
mainning points both forward and backward in time using affine
template matching. The backward-forward consistency check is
used to discard erroneous optical flow during tracking. Finally,
we check for the appearance consistency between patches of the
first and the last frame using Normalized Cross Correlation and
remove the entire trajectory if the score is below 0.8.

Trajectory classification: We exploit the fact that triangulation
based methods work for static points but produce large errors for
dynamic points in order to identify 2D trajectories of dynamic
points. This is done using these two heuristics: (1) the reprojection
error of a static point should be small regardless of which camera
frame it is triangulated from. We randomly sample frames along
the 2D trajectory to triangulate and consider the 2D trajectory
as belonging to a static point if the reprojection threshold is
smaller than 3 pixels for more than 80% of the time. (2) the
reprojection error of a dynamic point forms a steep valley as the
time offset passes by its true value. We reject any set of trajectories
as belonging to a dynamic point if the minimum of the cost valley
is not smaller than 80% of the average cost.

5.2 Sparse spatiotemporal bundle adjustment
We first evaluate the effect of properly modeling the rolling shutter
pose and its readout speed to the reconstruction. As shown in
Fig. 6, spatial modeling the rolling shutter produces significantly
more stable camera trajectory and lower reprojection error. As
showed in Tab. 2, although for artificially down sample frame rate
Fig. 5: Effect of accurate sub-frame alignment for the 3D trajectory estimation. (a) Point triangulation of frame accurate alignment gives large reconstruction error and creates different 3D shape with respect to other methods. (b) Incorrect sub-frame alignment generates 3D trajectory with many loops. (c) Trajectory estimated from correct sub-frame alignment is free from the loops. (d) Using DCT resampling for (c) gives smooth and shape preserving 3D trajectory.

Fig. 6: Analysis of the spatial camera calibration for the Checkerboard sequence with different camera models.

<table>
<thead>
<tr>
<th>Frame</th>
<th>No rolling shutter compensation</th>
<th>With rolling shutter compensation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static-Dynamic</td>
<td>No γ</td>
<td>With γ</td>
</tr>
<tr>
<td>Checkerboard</td>
<td>0.70</td>
<td>1.52</td>
</tr>
<tr>
<td>Jump</td>
<td>0.61</td>
<td>1.55</td>
</tr>
<tr>
<td>Dance</td>
<td>0.85</td>
<td>2.38</td>
</tr>
</tbody>
</table>

TABLE 2: Effect of modeling the rolling shutter readout on the reconstruction accuracy. While the temporal sequencing between cameras is still correct (due to the artificial down-sampling of the framerate), modeling γ results in smaller the reprojection error. Inaccurately reconstructed dynamic points also (slightly) negatively affect the reconstruction of static points.

videos, the rolling shutter readout becomes less significant given the artificially lengthen frame duration, the reconstructions with γ modeled are consistently more accurate. Theoretically, while the reconstruction of static points is not affected by γ, inaccurate reconstruction of dynamic points negatively affect the camera calibration parameters, which in turn affects the static points. For all results represented below, both the rolling shutter camera pose and readout speed are employed.

Tab.3 gives the complete quantitative evaluation on three video sequences in terms of (a) reprojection error in pixels for both stationary and dynamic points, (b) number and average length (time) of the 3D trajectories created using points from multiple views. Points with reprojection error exceeding the threshold are discarded. Noticeably, our proposed method produces several fold more trajectories, longer average trajectory length, and less reprojection error than geometry approach. For the checkerboard sequence, since the correspondences are known, its 3D points are intentionally not discarded. The resampling scheme consistently and noticeably further reduces re-projection error for all scenes. Similar to the analysis on synthetic data, the divide and conquer scheme is as accurate as the incremental reconstruction and alignment approach but is at least 10 times faster (40 times for Dance sequence).

**Checkerboard scene:** Since the ground truth location of the board is unknown, we quantify the reconstruction accuracy by measuring the deviation from the planar configuration for all the checkerboard corners. We reconstruct each corner independently. As depicted in Fig. 4, the reconstruction using geometric triangulation is at least 80 mm inaccurate. Conversely, most 3D corners estimated from our method have much smaller error (direct motion prior: 33 mm, DCT: 15 mm). Visually, the estimated trajectories using the method assemble themselves in the grid-like configuration of the physical board (see the supplementary video).

Fig. 5 shows the effect of accurate sub-frame alignment on the trajectory reconstruction. Due to the fast motion, geometry-based method produces trajectory with much different shape than the motion prior based method. We artificially alter the sub-frame of the offsets to create wrong frame sequencing between different cameras and optimize Eq. 3 for the trajectory. This results in trajectories with many small loops, a strong cue of incorrect alignment. Conversely, our reconstruction with correct time alignment is free from the loops. Our final result, obtained by DCT resampling, gives smooth and shape-preserving trajectories.

**Jump scene:** To visually evaluate the alignment, we scale the estimated offsets to show the alignment on the original footage at 120fps (see Fig. 7). Notice that the shadow cast by the folding cloth are well aligned across images. Fig. 8 shows our estimated trajectories for all methods. The point triangulation of frame-accurate alignment fails to reconstruct the fast action happening at the end of the action. Conversely, our method produces plausible metric reconstruction for the entire action even with relatively low frame-rate cameras. Due to the lack of ground truth data, we compare our reconstruction with the point triangulation using 120 fps videos, where few differences between the two reconstructions are observed.

**Dance scene:** We estimate per-frame camera intrinsic to account for the auto-focus function of smartphone cameras. Fig. 9 shows our trajectory reconstruction results. Our method reconstructs fast motion trajectories (jumping), longer and higher temporal resolution trajectories than point triangulation results at 15 fps. Since we discard many short 2D trajectories (thresholded at 10 samples), we reconstruct fewer 3D trajectories than geometric triangulation at 60 fps. However, the overall shape of the trajectories is similar.

Interestingly, this scene has a large number of static background points. This adversely reduces the spatial calibration accu-
TABLE 3: Reconstruction accuracy comparison between geometric triangulation and our proposed method. RMSE$^1$ and RMSE$^2$ are the results obtained by the incremental reconstruction and alignment and divide and conquer approaches, repetitively. The * denotes the results after resampling. Both approaches are equally accurate and the divide and conquer scheme is at least 10 times faster. Please see the text for more details.

### Accuracy Comparison

<table>
<thead>
<tr>
<th>Geometry</th>
<th>#Trajectory</th>
<th>Avg samples per trajectory</th>
<th>RMSE (pixels)</th>
<th>#Trajectory</th>
<th>Avg samples per trajectory</th>
<th>RMSE (pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Checkerboard</td>
<td>88</td>
<td>179.8</td>
<td>0.67</td>
<td>88</td>
<td>1023.0</td>
<td>0.67</td>
</tr>
<tr>
<td>Jump</td>
<td>717</td>
<td>36.4</td>
<td>0.59</td>
<td>3231</td>
<td>127.8</td>
<td>0.59</td>
</tr>
<tr>
<td>Dance</td>
<td>577</td>
<td>22.3</td>
<td>0.82</td>
<td>4105</td>
<td>216.4</td>
<td>0.82</td>
</tr>
</tbody>
</table>

### Acknowledgement

This research is supported by the NSF CNS-1446601, the ONR N00014-14-1-0595, and the Heinz Endowments “Plattform Pittsburgh”. Minh Vo is partly supported by the 2017 Qualcomm Innovation Fellowship. Part of this work is presented in [37].
Fig. 8: Jump scene. Point triangulation of frame-accurate alignment fails to reconstruct the fast action happened at the end of the sequence. Conversely, our motion prior based approach produces plausible reconstruction for the entire course of the action even with relatively low frame-rate cameras. Trajectories estimated from our approach highly resemble those generated by the frame-accurate alignment and triangulation at 120fps.

Fig. 9: Dance scene. The 3D trajectories are estimated using 10-15 fps cameras. Noticeably, the trajectories generated from frame accurate alignment and triangulation are fewer, shorter, and have lower temporal resolution than those reconstructed from motion prior based approaches.
Fig. 10: Evaluation of the spatiotemporal calibration. The blue and red lines are the estimated epipolar lines before and after spatiotemporal bundle adjustment, respectively. The epipolar lines estimated after spatiotemporal bundle adjustment have noticeable improvement at the foreground for cameras with a large number of visible dynamic points.

References

Fig. 11: The effect of subframe alignment for human shape fitting for the Dance and Jump scenes. Given the input frames (first column), due to fast motion, shape fitting assuming frame-level alignment and geometric constraint (second column) fails to match the silhouette of the person. Our motion prior based shape fitting to the 2D body joints (third column) better aligns the observed silhouette but fails when the joints are not visible. Adding the silhouette constraint to the motion prior based fitting produces the best results (last column).
Fig. 12: Human mesh fitting for different person in the Dance sequence (top and bottom left) and Jump sequence (bottom right). The color encodes the relative action time.

Minh Vo is a Research Scientist at Facebook Reality Lab in Sausalito and a PhD student in The Robotics Institute, Carnegie Mellon University. He received his MS degree in Robotics from Carnegie Mellon University in 2015. His research focuses on large scale dynamic scene reconstruction in the wild. He is the recipient of the 2017 Qualcomm Innovation Fellowship.

Yaser Sheikh is the Director of the Facebook Reality Lab in Pittsburgh and is an Associate Professor at the Robotics Institute at Carnegie Mellon University. His research is broadly focused on machine perception of social behavior, spanning computer vision, computer graphics, and machine learning. With colleagues, he has won Popular Sciences Best of What’s New Award, the Honda Initiation Award (2010), best student paper award at CVPR (2018), best paper awards at WACV (2012), SAP (2012), SCA (2010), ICCV THEMIS (2009), best demo award at ECCV (2016), and placed first in the MSCOCO Keypoint Challenge (2016); he has also
received the Hillman Fellowship for Excellence in Computer Science Research (2004). Yaser has served as a senior committee member at leading conferences in computer vision, computer graphics, and robotics including SIGGRAPH (2013, 2014), CVPR (2014, 2015, 2018), ICRA (2014, 2016), ICCP (2011), and served as an Associate Editor of CVIU. His research is sponsored by various government research offices, including NSF and DARPA, and several industrial partners including the Intel Corporation, the Walt Disney Company, Nissan, Honda, Toyota, and the Samsung Group. His research has been featured by various media outlets including The New York Times, The Verge, Popular Science, BBC, MSNBC, New Scientist, slashdot, and WIRED.

Srinivasa G. Narasimhan is a Professor of Robotics and ECE (by courtesy) at Carnegie Mellon University. He obtained his PhD from Columbia University in 2003. His group focuses on novel techniques for imaging, illumination and light transport to enable applications in vision, graphics, robotics, agriculture and medical imaging. His works have received several awards: Best Paper Award (CVPR 2019), Best Demo Award (ICCP 2015), A9 Best Demo Award (CVPR 2015), Marr Prize Honorable Mention Award (2013), FORD URP Award (2013), Best Paper Runner up Prize (ACM IS3D 2013), Best Paper Honorable Mention Award (ICCP 2012), Best Paper Award (IEEE PROCAMS 2009), the Okawa Research Grant (2009), the NSF CAREER Award (2007), Adobe Best Paper Award (IEEE Workshop on Physics-based methods in computer vision, ICCV 2007) and Best Paper Honorable Mention Award (CVPR 2000). His research has been covered in popular press including NY Times, BBC, PC magazine and IEEE Spectrum and is highlighted by NSF and NAE. He is the co-inventor of programmable headlights, Aqualux 3D display, Assorted-pixels, Motion-aware cameras, Episcan3D, EpiToF3D, and programmable triangulation light curtains. He co-chaired the International Symposium on Volumetric Scattering in Vision and Graphics in 2007, the IEEE Workshop on Projector-Camera Systems (PROCAMS) in 2010, and the IEEE International Conference on Computational Photography (ICCP) in 2011, co-edited a special journal issue on Computational Photography, and serves on the editorial board of the International Journal of Computer Vision and as Area Chair of top computer vision conferences (CVPR, ICCV, ECCV, BMVC, ACCV).