Spatiotemporal Bundle Adjustment for Dynamic 3D Reconstruction *

Minh Vo Srinivasa G. Narasimhan Yaser Sheikh
Carnegie Mellon University
{mpvo, srinivas, yaser}@cs.cmu.edu

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 purposed to estimate the temporal alignment between cameras. In this paper, we present a spatiotemporal bundle adjustment approach 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. We present an end-to-end pipeline that takes multiple uncalibrated and unsynchronized video streams and produces a dynamic reconstruction of the event. Because the videos are aligned with sub-frame precision, we reconstruct 3D trajectories of unconstrained outdoor activities at much higher temporal resolution than the input videos.*

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 [11], and the algorithmic edifice of bundle adjustment [19] 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 [10]. 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 [13], 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[1] [14, 8, 21].

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 [2], 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 [2]; smooth motion [13, 20]; sparsity priors [27]; low rank spatiotemporal priors [15]. Estimating the relative temporal offsets of videos captured by the moving cameras is more involved [24, 9]. Currently, the most stable temporal alignment methods require corresponding 2D trajectories as input [23, 4, 12, 5, 18] 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 [3, 26].

In this paper, we introduce the concept of spatiotemporal bundle adjustment that jointly optimizes for all for sub-

*http://www.cs.cmu.edu/ ILIM/projects/IM/STBA/

[1]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 näive 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.

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. 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 solve 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. Based on this framework, we present an end-to-end pipeline that takes multiple uncalibrated and unsynchronized videos and outputs a dynamic 3D reconstruction of the scene. This pipeline, inspired by the large-scale static scene reconstruction [16, 7], is a step towards dynamic event reconstruction in the wild. As a demonstration, we reconstruct 3D trajectories of dynamic actions captured outdoor by ten smartphones without any constraints.

2. Problem Formulation

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 $x_c^n(f)$ on camera c at frame f is given by:

$$\begin{bmatrix} x_c^n(f) \\ 1 \end{bmatrix} \equiv K_c(f) \begin{bmatrix} R_c(f) & T_c(f) \end{bmatrix} \begin{bmatrix} X^n(t) \\ 1 \end{bmatrix}, \quad (1)$$

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 $x_c^n(f) = P_c(f, X^n(t))$. The time corresponding to frame f is related to the continuous global time t linearly: $f = \alpha_c t + \beta_c$, where α_c and β_c are the camera frame rate and time offset. For a static 3D point, $X^n(t)$ is a constant.

Image reprojection cost. Regardless of its motion, the reconstruction of a 3D point must satisfy Eq. 1. This gives the standard reprojection error, which we accumulate over all 2D points observed by all C cameras for all frames F_c:

$$S_I = \sum_{c=1}^{C} \sum_{n=1}^{N} \sum_{f=1}^{F_c} V_c^n(f) \sigma_c^n(f) \| P_c(f, X^n(t)) - x_c^n(f) \|^2 \quad (2)$$

where, S_I is the image reprojection cost, $V_c^n(f)$ is a binary indicator of the point-camera visibility, and $\sigma_c^n(f)$ is a scalar, weighting the contribution of $x_c^n(f)$ to S_I. Since the localization uncertainty of an image point $x_c^n(f)$ is proportional to its scale [25], we use the inverse of the feature scale as the weighting term for each residual term in S_I.

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:

$$S = \arg \min_{\mathbf{X}(t), \{\mathbf{K}, \mathbf{R}, \mathbf{t}\}, \boldsymbol{\alpha}, \boldsymbol{\beta}} S_I + S_M. \quad (3)$$

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

3. 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.

3.1. 3D Trajectory Motion Priors

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 [6] (see Fig. 1 for the formal definitions). In each of these priors, m denotes the mass of the 3D point, g is the gravitational acceleration force acting on the point at height $h(t)$,

[2]We actually use the square of the resulting forces.

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 entire operating time, 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 error in the 3D trajectory, either by inaccurate estimation of points along the trajectory or wrong temporal sequencing between points observed across different cameras, generates higher motion prior cost.

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

$$S_M = \sum_{n=1}^{N} \sum_{i=1}^{T^n-1} w_n(t) \frac{m_n}{2} v_n(t^i)^2 (t^{i+1} - t^i), \quad (4)$$

where $\gamma_n(t)$ is the weighting scalar and m_n is the point mass, assumed be to identical for all 3D points and set to be 1. We approximate the instantaneous speed $v(t^i)$ 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 ϵ to the denominator to avoid instability caused by 3D points observed at approximately same time. Eq. 4 is rewritten as:

$$S_M = \sum_{n=1}^{N} \sum_{i=0}^{T^n-1} \frac{w_n(t)}{2} \left\| \frac{X^n(t^{i+1}) - X^n(t^i)}{t^{i+1} - t^i + \epsilon} \right\|^2 (t^{i+1} - t^i), \quad (5)$$

Using the uncertainty $\sigma_c^n(f)$ 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 λ and its scale μ, relating the focal length to the physical pixel size, as $w_n = \sigma_c^n \mu \lambda$. The least force and least action prior costs can be computed similarly.

3.2. Evaluation on 3D 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 data containing over 2500 sequences of common human activities such as playing, sitting, dancing, running and jumping, captured at 120 fps.

Input: $\{\mathbf{x}_c(t)\}, \{\mathbf{K}', \mathbf{R}', \mathbf{T}'\}, \boldsymbol{\beta}'$
Output: $\{\mathbf{X}(t)_p\}, \{\mathbf{K}, \mathbf{R}, \mathbf{T}\}, \boldsymbol{\beta}$
1. Refine the alignment pairwise (Sec. 4.1.1)
2. Generate prioritized camera list (Sec. 4.1.2)
3. while *All cameras haved **NOT** been processed* **do**
  for *All cameras slots* **do**
    Solve Eq. 3 for $\{\mathbf{X}_p(t)\}$ and $\boldsymbol{\beta}$
    if *No sequencing flipped* **then**
      Record the STBA cost and its solution.
    else
      Discard the solution;
    end
  end
  Accept the solution with the smallest cost
end
(Sec. 4.1.3)
4. Solve Eq. 3 for $\{\mathbf{X}(t)_p\}, \{\mathbf{K}, \mathbf{R}, \mathbf{T}\}, \boldsymbol{\beta}$ (Sec. 4.2)
5. Trajectory resampling on $\{\mathbf{X}(t)_p\}$ (Sec. 4.2)

Algorithm 1: Spatiotemporal bundle adjustment

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 and both estimate the time offset between the two trajectories well for low noise levels. When more noise is added to the trajectory sequences, 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 (Fig. 4). The least action prior, on the other hand, gives biased results even when no noise is added to the 3D data.

Since the alignment results using the least kinetic energy prior is similar to the least force prior, we only present our algorithm for the least kinetic prior in the remainder of the paper. Extension to the least force is straight forward.

4. Spatiotemporal Bundle Adjustment

Unlike traditional bundle adjustment, 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 and (b) motion prior cost is strongly discontinuous as small

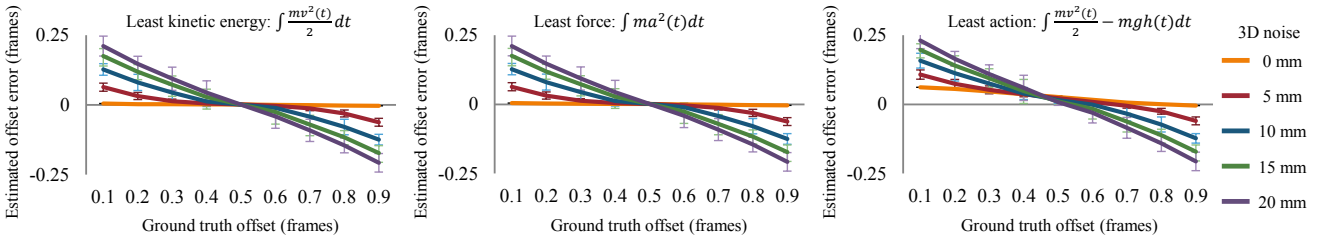


Figure 1: Evaluation of the motion priors on 3D motion capture data. The least kinetic energy prior and least force prior performs similarly and both estimate the time offset between two noisy sequences obtained by uniformly sampling a 3D trajectory from different starting times. The least action prior gives biased results even for the no-noise case.

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.

To solve this problem, we first follow an incremental reconstruction and alignment approach, where cameras are added one at a time. In addition to being more computationally efficient (as in traditional bundle adjustment), we show that this approach allows us to enforce the motion prior constraint strictly without any discontinuities due to incorrect time ordering of cameras. This allows us to then use LM optimization to jointly estimate all the spatiotemporal camera parameters, and static points and dynamic trajectories.

We start with initial estimates of all the quantities using a geometry (or triangulation) based method [16, 5]. Even though the triangulation constraint is not strictly satisfied, the estimates provide a good starting point for the incremental reconstruction and alignment. The entire method is summarized in Algorithm.1.

4.1. Incremental Reconstruction and Alignment

4.1.1 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.

4.1.2 Which camera to add next?

As in incremental SfM [16, 7], 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

$$E_{ij} = \sum_{k=1, k\neq i,j}^{C} S_{ij} \frac{|t_{ij} + t_{jk} - t_{ik}|}{N_{ij} B_{ij}}, \tag{6}$$

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 [5, 22], 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.

4.1.3 Estimating the time offset of the next camera

We temporally order the current 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 trial 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.

4.2. Final Optimization and DCT Resampling

Starting from the results of the above incremental procedure, we now jointly optimize Eq. 3 for all the camera parameters, 3D points and trajectories without allowing any reordering of the cameras.

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. Thus, as a post-processing step, we fit the weighted complete DCT basis function to the estimated trajectories. Our use of DCT for resampling is exactly equivalent to our sample-wise motion prior [17] and is not an extra smoothing prior. For the uniform DCT resampling, the least kinetic energy prior cost can be rewritten as:

$$S'_M = \sum_{n=1}^{N} E^{n\top} W^n E^n \Delta t, \tag{7}$$

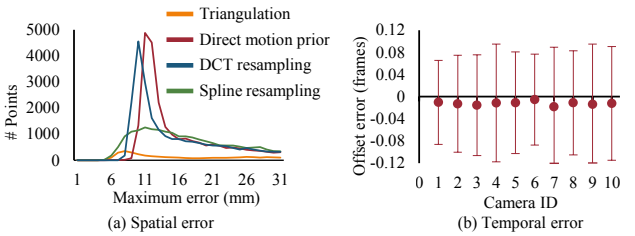


Figure 2: Evaluation of the motion priors on the Motion Capture database for simultaneous 3D reconstruction and sub-frame temporal alignment. (a) Spatially, the trajectories estimated using the motion prior achieves higher accuracy than generic B-spline trajectories basis. Frame level alignment geometric triangulation spreads the error to all cameras and estimates less accurate 3D trajectories. (b) Temporally, our motion prior based method estimates the time offset between cameras with sub-frame accuracy.

where E^n is the DCT coefficient of the 3D trajectory n, W^n is a predefined diagonal matrix, weighting the contribution of the bases, and Δt is the resampling period. The 3D trajectory $X^n(t)$ is related to E^n by $X^n(t) = B^{n\top} E^n$, where B^n is a predefined DCT basis matrix. The dimension of B^n and W^n depend on the trajectory length. We replace the trajectory $X^n(t)$ by $B^{n\top} E^n$ and rewrite Eq. 3 as:

$$S = \underset{E}{\operatorname{argmin}}\, \lambda_1 S'_I + \lambda_2 S'_M, \tag{8}$$

where λ_1 and λ_2 are the weighting scalars and S'_I is the re-projection error computed using the resampled trajectories. While applying resampling to the incremental reconstruction loop can improve the 3D trajectories and the temporal alignment, it requires inverting a large and dense matrix of the DCT coefficients, which is computationally demanding. Thus, we only use this scheme as a post-processing.

5. Analysis on Mocap Data

We validate our approach on synthetic data generated from the CMU Motion Capture database. The ground truth trajectory, captured at 120 fps, is imaged by 10 perspective cameras with resolution of 1920x1080 and 12fps. All cameras are uniformly arranged in 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 and none of them generates cameras observing the 3D points synchronously. We assume that the initial offsets are frame 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 2D trajectories.

The reconstruction and alignment errors are summarized in Fig.2 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 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.

	Geometry	Spline	Motion prior (after 3)	Motion prior (after 5)
Static	3.45	2.93	2.54	2.41
Dynamic	17.8	1.68	0.85	0.74

Table 1: The reprojection error for the entire CMU Mocap dataset. The results for motion prior based method are shown for different stages of Algo. 1.

6. Analysis on Handheld Camera Videos

We develop an end-to-end system that takes video streams for multiple temporally unaligned and spatially uncalibrated cameras and produces the spatiotemporal calibration parameters as well as the 3D static points and dynamic trajectories. We show the results for 3 scenes: checkerboard, jump, and dance, captured by either smartphone or GoPro cameras, which are rolling shutter camera[3]. We quantify the error in 3D trajectory estimation and effect of sub-frame alignment using the Checkerboard sequence. The Jump sequence demonstrates our ability to handle fast motion using low framerate cameras. The Dance scene showcases the situation where the static background and dynamic foreground are separated by a large distance.

Table 2 presents complete quantitative evaluation on three video sequences in terms of (a) re-projection 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. Noticeably, our method estimate several fold more trajectories and the longer average trajectory length than geometry approach. For the checkerboard sequence, since the correspondences are known, its 3D points are intentionally not discarded. We also estimate the dynamic points with less re-projection error, especially for fast actions. The jump in the Jump-sequence is not reconstructed at all (see Fig. 6) by the geometry-based method and the low average re-projection error (1.91) is due to the slow initial motion of the person. Lastly, optimizing for the camera pose along with the resampling scheme consistently yields a further noticeably smaller re-projection error for the Dance sequence with large background-foreground separation.

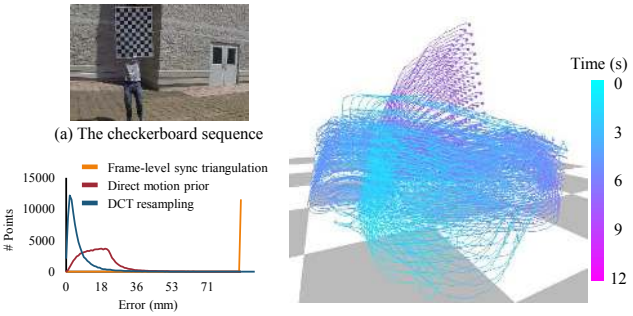
[3]Refer the supplementary material for the data reproprecessing

	Geometry				Motion Prior					
	#Trajectory	Avg samples per trajectory	RMSE (pixels) Static—Dynamic		#Trajectory	Avg samples per trajectory	RMSE (pixels) Static—Dynamic		RMSE* (pixels) Static—Dynamic	
Checkerboard	88	179.8	0.67	6.59	88	1023.0	0.67	1.21	0.65	1.15
Jump	717	36.4	0.59	1.91	3231	127.8	0.59	1.34	0.6	1.26
Dance	577	22.3	0.82	5.23	4105	216.4	0.82	2.12	0.85	1.71

Table 2: Quantitative analysis for the outdoor sequences. RMSE and RMSE* are the results after stage 3 and 5 of Algo. 1.

(a) Geometric triangulation @10fps (b) Motion prior with incorrect sub-frame alignment @10fps (c) Motion prior with correct sub-frame alignment @10fps (d) DCT resampling of (c)

Figure 4: 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.



(a) The checkerboard sequence (c) Error histogram of the reconstructed checkerboard (b) The 3D trajectory of the checkerboard corners

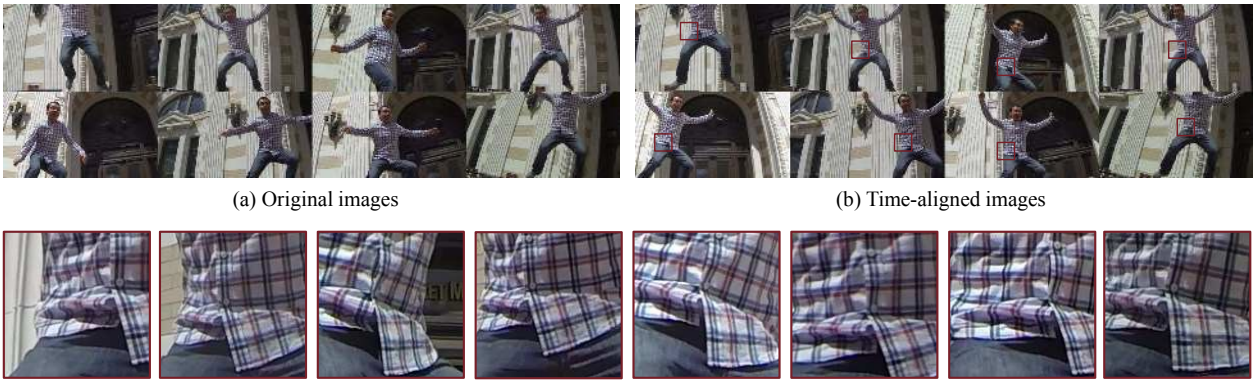
Figure 3: Accuracy evaluation of the checkerboard corner 3D trajectories. While the reconstruction is conducted independently at every corner, collectively, the estimated 3D trajectories assemble themselves in the grid-like configuration. Our methods produce trajectories with significantly smaller error than naive geometric triangulation.

Checkerboard scene: This scene is captured by 7 GoPro cameras with resolution of 1920x1080 at 60fps. We down sample all videos to 10fps to mimic faster motion. We rigidly align the ground truth configuration of the checkerboard to its estimated position and compute their difference for every corner. While we applied our method independently to each checkerboard corner, collectively, the estimated trajectories assemble themselves in the grid-like configuration of the physical board (see Fig. 3). Quantitatively, point triangulation of frame accurate alignment produces error of at least 80 mm for every corner. Conversely, most 3D corners estimated from our method have much smaller error (direct motion prior: 35 mm, DCT: 18 mm).

Fig. 4 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: This scene is captured by 8 GoPro cameras at 120 fps at 1280x720 resolution. We compute 2D trajectories at 120 fps and artificially down sample them 30fps to mimic faster motion. To evaluate the alignment, we increase the estimated offsets by 4 times and show the alignment on the original footage at 120fps (see Fig. 5). Notice that the shadow cast by the folding cloth are well aligned across images. This means that our alignment is at least 0.25 frames accurate using 30 fps data.

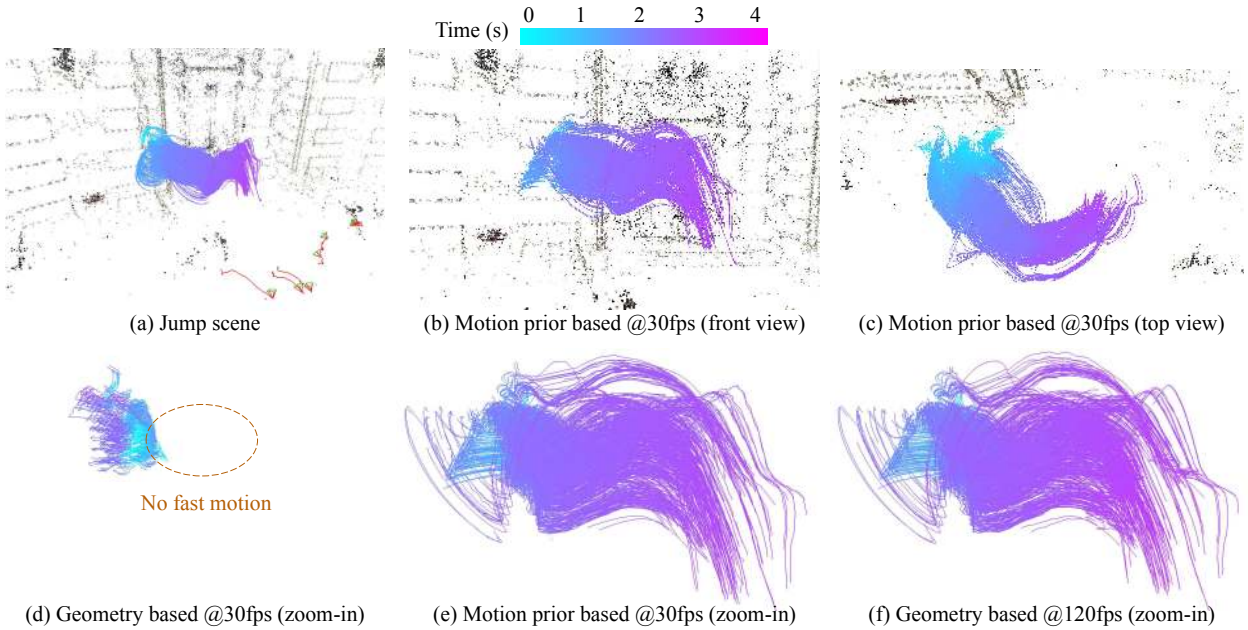
Fig. 6 shows our estimated trajectories for all methods. While the point triangulation of frame accurate alignment fails to reconstruct the fast action happening at the end of the action, our method produces plausible metric reconstruction for the entire action even with relatively low frame-rate cameras. Due to the lack of ground truth data, we compare the our reconstruction with the point triangulation using 120 fps videos, where few differences are seen



(a) Original images (b) Time-aligned images

(c) Insets of the time-aligned images

Figure 5: Temporal alignment. (a) Original unaligned images. (b) Our aligned images, estimated from temporally down sampled video at 30 fps, are shown for the original video captured at 120 fps. (c) Inset of aligned images. The shadow casted by the folding cloth are well temporally aligned across images.



(a) Jump scene (b) Motion prior based @30fps (front view) (c) Motion prior based @30fps (top view)

(d) Geometry based @30fps (zoom-in) (e) Motion prior based @30fps (zoom-in) (f) Geometry based @120fps (zoom-in)

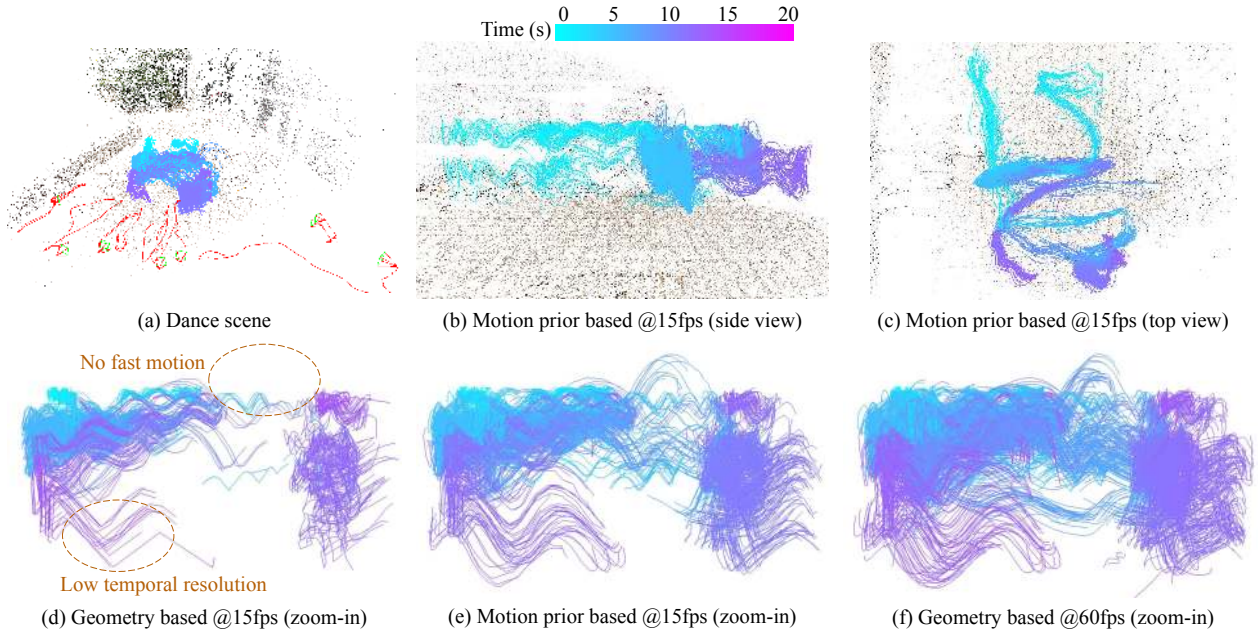
Figure 6: Jump scene. Point triangulation of frame accurate alignment fails to reconstruct the fast action happened at the end of the sequence. Conversely, our motion prior based approach produces plausible reconstruction for the entire course of the action even with relatively low frame-rate cameras. Trajectories estimated from our approach highly resembles those generated by the frame accurate alignment and triangulation at 120fps.

between the two reconstructions.

Dance scene: This scene is captured by five iPhone 6 and five Samsung Galaxy 6 at 60fps. As before, we track points at 60fps and down sample them to 15fps for processing. We estimate per-frame camera intrinsic to account for the auto-focus function of smartphone cameras.

Fig. 7 shows our trajectory reconstruction results. Our method reconstructs fast motion trajectories (jumping), longer and higher temporal resolution trajectories than point triangulation results at 15fps. Since we discard many short 2D trajectories (thresholded at 10 samples), we reconstructs fewer 3D trajectories than geometric triangulation at 60fps. However, the overall shape of the trajectories are similar.

Interestingly, this scene has a large number of static background points. This adversely reduces the spatial calibration accuracy for the foreground points (see Fig. 8). Using our algorithm clearly improves the spatial calibration for cameras with enough number of visible dynamic points.

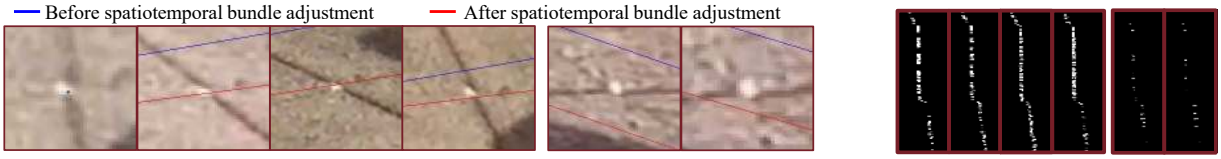


(a) Dance scene (b) Motion prior based @15fps (side view) (c) Motion prior based @15fps (top view)

(d) Geometry based @15fps (zoom-in) (e) Motion prior based @15fps (zoom-in) (f) Geometry based @60fps (zoom-in)

Figure 7: 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.

(a) A subset of the temporally aligned images and their epipolar lines corresponding to the point in the first camera image



(b) Insets of the (a): Success (left) and failure (right) epipolar estimation. (c) Visibility matrix of the dynamic points

Figure 8: 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.

7. Discussion

While our incremental reconstruction and alignment can strictly enforce the motion prior, linearly searching for the best sequencing time slot followed by a local optimization is computational demanding. As the number of camera increases, the number of slots increases. The number of trajectory samples also rises. Thus, the computational complexity increases every iteration.

One of the our biggest obstacles is the requirement of corresponding 2D trajectories across cameras. Just as SIFT descriptor for point matching has revolutionized static scene reconstruction, trajectory descriptor is needed for dynamic scene reconstruction. To disambiguate the matching, such a descriptor must accumulate information spatiotemporally. Further effort must be invested to solve this problem.

Acknowledgement

This research is supported by the NSF CNS-1446601, the ONR N00014-14-1-0595, and the Heinz Endowments "Plattform Pittsburgh".

References

[1] http://mocap.cs.cmu.edu.

[2] S. Avidan and A. Shashua. Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence. *PAMI*, 2000.

[3] T. Basha, Y. Moses, and S. Avidan. Photo sequencing. In *ECCV 2012*, pages 654–667. Springer, 2012.

[4] Y. Caspi, D. Simakov, and M. Irani. Feature-based sequence-to-sequence matching. *IJCV*, 2006.

[5] A. Elhayek, C. Stoll, K. Kim, H. Seidel, and C. Theobalt. Feature-based multi-video synchronization with subframe accuracy. *Pattern Recognition*, 2012.

[6] R. P. Feynman, R. B. Leighton, and M. Sands. The Feynman lectures in physics, Mainly Electromagnetis and Matter, Vol. ii, 1963.

[7] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, et al. Building rome on a cloudless day. In *ECCV*. 2010.

[8] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *PAMI*, 2010.

[9] T. Gaspar, P. Oliveira, and P. Favaro. Synchronization of two independently moving cameras without feature correspondences. In *ECCV*. 2014.

[10] R. Latimer, J. Holloway, A. Veeraraghavan, and A. Sabharwal. Socialsync: Sub-frame synchronization in a smartphone camera network. In *ECCV*. Springer, 2014.

[11] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 1981.

[12] F. L. Pádua, R. L. Carceroni, G. A. Santos, and K. N. Kutulakos. Linear sequence-to-sequence alignment. *PAMI*, 2010.

[13] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh. 3d trajectory reconstruction under perspective projection. *IJCV*, 2015.

[14] K. Raguse and C. Heipke. Photogrammetric synchronization of image sequences. In *Proc. of the ISPRS Commission V Symp. on Image Eng. and Vision Metrology*, 2006.

[15] T. Simon, J. Valmadre, I. Matthews, and Y. Sheikh. Separable spatiotemporal priors for convex reconstruction of time-varying 3d point clouds. In *ECCV*. 2014.

[16] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *SIGGRAPH*. ACM, 2006.

[17] G. Strang. The discrete cosine transform. *SIAM review*, 1999.

[18] P. A. Tresadern and I. D. Reid. Video synchronization from human motion using rank constraints. *CVIU*, 2009.

[19] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment A modern synthesis. In *Vision algorithms: theory and practice*. 2000.

[20] J. Valmadre and S. Lucey. General trajectory prior for non-rigid reconstruction. In *CVPR*, 2012.

[21] M. Vo, Z. Wang, B. Pan, and T. Pan. Hyper-accurate flexible calibration technique for fringe-projection-based three-dimensional imaging. *Optics Express*, 2012.

[22] O. Wang, C. Schroers, H. Zimmer, M. Gross, and A. Sorkine-Hornung. Videosnapping: interactive synchronization of multiple videos. *SIGGRAPH*, 2014.

[23] D. Wedge, D. Huynh, and P. Kovesi. Motion guided video sequence synchronization. In *ACCV*. 2006.

[24] J. Yan and M. Pollefeys. Video synchronization via space-time interest point distribution. In *ACIVS*, 2004.

[25] B. Zeisl, P. F. Georgel, F. Schweiger, E. G. Steinbach, and N. Navab. Estimation of location uncertainty for scale invariant features points. In *BMVC*, 2009.

[26] E. Zheng, D. Ji, E. Dunn, and J.-M. Frahm. Jsparse dynamic 3D reconstruction from unsynchronized videos. In *ICCV*. 2015.

[27] Y. Zhu and S. Lucey. Convolutional sparse coding for trajectory reconstruction. *PAMI*, 2015.