OffBeat: Multicoset Sampling for 4D Capture Beyond the Camera Frame Rate

Chi-Wei Lee An-Kai Chen Cheng-Han Chiang Yi-Chun Chang Po-Chih Kuo
National Tsing Hua University, Hsinchu, Taiwan

arthur031221.github.io/offbeat

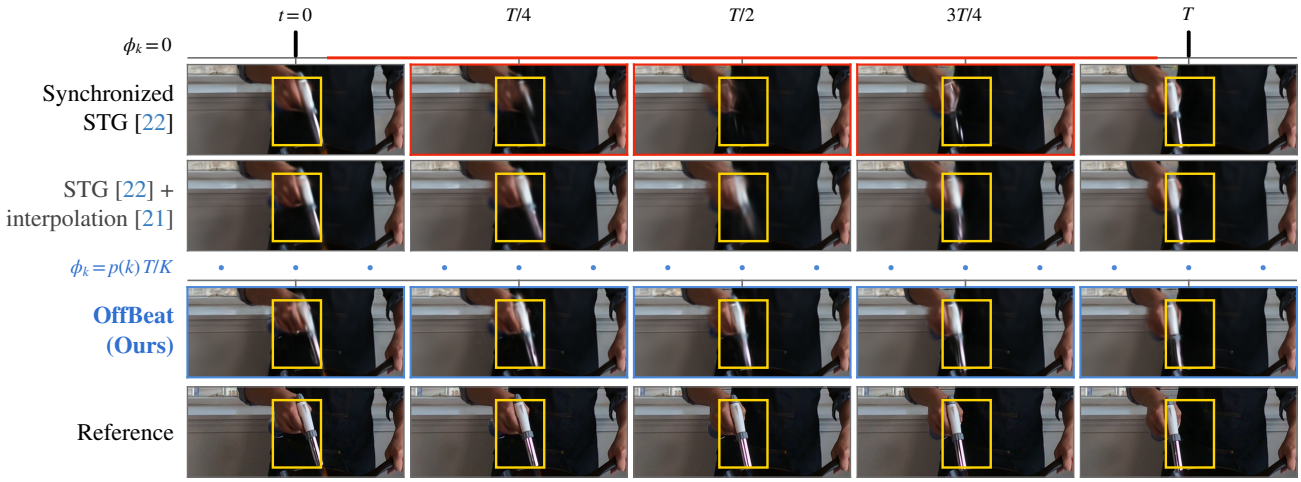


Figure 1. **With the same cameras and frame count, staggered trigger times let OffBeat follow the motion between synchronized frames.** Each of the 18 training cameras of N3DV Flame Salmon keeps five frames of a 2 s clip. Test-view renders over one camera frame interval T and the trigger offsets ϕ_k of both rigs, where camera k is in group $p(k)$ of K=12. Red: instants the synchronized rig misses, where synchronized STG fades the torch and interpolation blurs it. Yellow boxes: most dynamic region. STG: Spacetime Gaussians.

Abstract

For 4D reconstruction, a synchronized multi-camera rig records all views at the same timestamps and misses the motion between them. Staggered triggers spread the same frames over more timestamps. However, when the cameras along each row fire one after another, as in 4DSloMo's schedule on the Neural 3D Video (N3DV) rig, the row acts as one moving camera that confounds depth with motion. We introduce OffBeat, which models a staggered rig as a spatially distributed multicoset sampler: shifted copies of one sampling grid. From the calibration alone, OffBeat derives camera groups that fire together and their firing order. Since one group records each timestamp, a short-lived primitive can fit an error its cameras share, such as a color offset, and flicker in novel views. We show that a primitive's opacity over time weights all groups equally, for every number of groups, only if it lasts at least one camera frame interval. On Spacetime Gaussians, a minimum lifetime approximates this bound, and a closed-form smoothed box of that length satisfies it exactly, with less flicker than a Gaussian in its place at nearly equal PSNR. In simulated staggered capture on N3DV, OffBeat outperforms the full 4DSloMo pipeline with video diffusion on every test frame, by 3.04 dB in mean dynamic-region PSNR. Between the first and last synchronized frame, it outperforms the best synchronized method of each scene, with or without frame interpolation, by 1.98 dB in mean dynamic-region PSNR, while synchronized capture is better at its own frames. On PanopticSports, in this frame range, OffBeat is comparable in PSNR to synchronized capture with interpolation. On every N3DV scene, replacing 4DSloMo's schedule with ours raises the dynamic-region PSNR of Spacetime Gaussians, 4DSloMo without diffusion, and our reconstruction. Multi-camera rigs could follow faster motion at the same frame count by changing only when each camera fires.

Emails: {levi74108520963, kylechen.ankai, faidavid7, changg587}@gmail.com, kuopc@cs.nthu.edu.tw

1. Introduction

4D reconstruction from multi-view video [20, 22, 47] still fails on fast non-rigid motion, such as a flame, a hand turning food, or a ball leaving a hand. One cause is temporal sampling: in a synchronized rig, all N cameras record at the same timestamps, once every camera frame interval T. Adding cameras adds views of these timestamps and leaves the temporal bandwidth at $1/(2T)$. A higher frame rate costs resolution, exposure, storage and link bandwidth.

Staggered triggers fire the cameras in K groups spaced T/K apart, which spreads the same number of frames over K times as many timestamps (Fig. 1). For nearly co-located cameras this is the classic high-speed array [1, 37, 46], and in a dense array staggered triggers sampled the video light field better than synchronized ones [46]. 4DSloMo [9] staggers a multi-view rig to raise its effective frame rate and repairs sparse-view artifacts at each timestamp with a 14B-parameter video diffusion model.

On the Neural 3D Video rig (N3DV) [20], the modular schedule of 4DSloMo, which gives camera k the offset $(k \bmod K)\,T/K$, fires each of the two camera rows in position order. Within a camera frame interval, such a row acts as one camera moving at constant speed: regardless of row size, a static point is indistinguishable from a moving point at another depth (Sec. 3.2, Fig. 2a).

We model a staggered rig as a spatially distributed multicoset sampler [28, 41]: cameras that fire together form a *trigger group* and sample at the camera frame rate on a shifted copy of one grid. As in multichannel sampling, a point whose motion is band-limited to half the effective frame rate K/T is identifiable if every group triangulates it (Sec. 3.1). Single-camera groups, as on N3DV, cannot triangulate; a motion model over one camera frame interval pools them. From the calibration alone, we choose the groups by how well the others triangulate, and their order by the predictive variance of novel views (Sec. 3.2).

The reconstruction faces a second problem: as each timestamp is recorded by a single group, an error its cameras share, such as a color or calibration offset, is fit exactly by spurious content present only at those timestamps, which a novel view sees as flicker at the camera frame rate (Sec. 3.3). A primitive can supply such content in every period only if its temporal opacity window, summed over copies one period apart, weights the groups unequally. We show that the windows weighting all groups equally for any number of groups and at any center are exactly the convolutions of a box of length T (Prop. 1). Hence a primitive's lifetime (its window's duration) is at least T rather than the trigger spacing T/K, while its motion is sampled at the effective frame rate. On Spacetime Gaussians (STG) [22], a *lifetime floor* approximates this bound, and a closed-form *period window*, a smoothed box, satisfies it exactly (Sec. 3.4).

In summary, we make the following contributions:

- We introduce OffBeat, which models a staggered rig as a spatially distributed multicoset sampler, and show that a position-ordered row confounds depth with motion.
- From the calibration alone, we derive which cameras fire together and in which order.
- We show that temporal windows weighting all groups equally for every K must last at least T, and derive a closed-form period window that satisfies this bound.
- In simulated staggered capture on N3DV, OffBeat outperforms the full 4DSloMo pipeline and follows motion between synchronized frames better on average than a synchronized rig of equal camera frame rate, which is better at its own frames. On PanopticSports, up to the last synchronized frame, OffBeat is comparable in PSNR to synchronized capture with interpolation.

2. Related Work

Staggered and unsynchronized capture. In high-speed arrays of nearly co-located cameras [1, 37, 46], the firing order is a design parameter [46]. Color-coded illumination yields high-speed 3D under controlled light [27]. Methods for unsynchronized video treat the time offsets as nuisances under smoothness priors [3, 29, 42, 52], estimate them [2, 8, 15, 19, 48, 53], or recover structure and motion from asynchronous tracks [11, 39]. 4DSloMo [9] staggers a rig with a modular schedule.

Multichannel sampling. In multichannel sampling, K channels at $1/K$ of the Nyquist rate recover a band-limited signal [28, 41]. Parallel MRI shares this structure [30, 40], and CAIPIRINHA [7] shifts the sampling pattern such that aliases fall where the coil sensitivities differ. Our ordering criterion plays this role for cameras by decorrelating the trigger offsets from the camera positions.

4D Gaussians and sampling-aware representations. Dynamic extensions of Gaussian splatting [13] use deformation fields [47], native 4D primitives [49], per-Gaussian trajectories with temporal opacity [18, 22, 45], or tracked Gaussians [24]. RetimeGS [44] and SPIN-4DGS [16] target temporal aliasing and large displacements between synchronized frames. Mip-Splatting [50] bounds the 3D frequency of a primitive by the sampling rate of the training views, and our lifetime bound is its temporal counterpart. SharpTimeGS [23] learns a flat-top lifespan for each primitive to balance static and dynamic content, whereas our window's box has the derived length T.

3. Method

3.1. Staggered rigs as multicoset samplers

Each of the N training cameras records a frame once every camera frame interval T. Camera k fires at its trigger offset

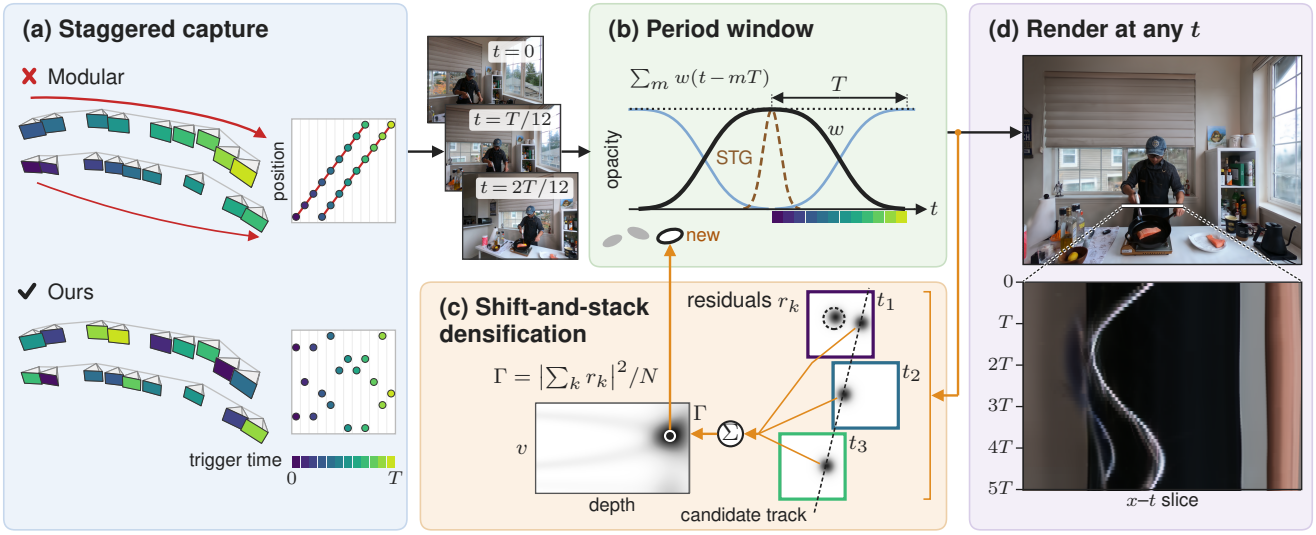


Figure 2. **OffBeat overview.** Colors are trigger times within the camera frame interval T. (a) The modular schedule fires each camera row in position order, so the row acts as one moving camera, confounding depth with motion; ours keeps them separable. (b) Each Gaussian's opacity follows a period window w whose copies T apart (blue) sum to a constant, weighting all groups equally; STG's shorter Gaussian window (dashed) can fit one group's error. (c) Residuals r_k summed along a candidate track give a depth–velocity score Γ that ranks a single-camera error (circled) low; new Gaussians are added at the maxima of Γ. (d) Our test-view render and x–t slice.

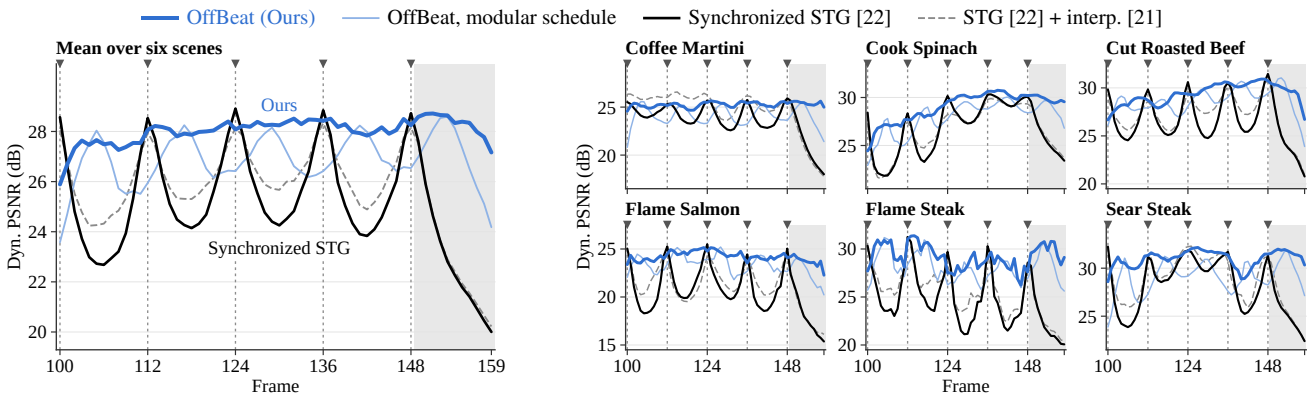


Figure 3. **Dynamic-region PSNR per frame on N3DV.** Left: mean over the six scenes. Right: each scene. Synchronized capture drops between its frames (▼) and after the last one (gray band), while OffBeat varies much less except near the first and last test frames.

ϕ_k, and a synchronized rig has all offsets zero. A K-group stagger fires the cameras S_p of group p at the offset pT/K. All triggers form the *joint grid* of spacing T/K, whose rate K/T is the effective frame rate of the rig [9]. Each instant of this grid is recorded by one group alone.

Camera k observes a scene point x through its 2×3 projection Jacobian D_k, with Fisher information $G_k = D_k^\top D_k/\sigma^2$ for pixel noise σ. G_k has rank two, and its null space is the viewing ray ℓ_k. Group p has the information $H_p = \sum_{k \in S_p} G_k$. A single group cannot separate frequencies that differ by a multiple of the camera frame rate. The K such frequencies inside the band of the joint grid form an *alias set*, and the Fourier coefficients of $x(t)$ there are its *alias components*; each group observes its own phase-weighted sum of them. For a trajectory band-limited to the Nyquist frequency of the joint grid, the Fisher information of its Fourier coefficients is block-diagonal over alias sets, with blocks built from the H_p, as in multichannel sampling [28, 30].

We prove two properties of this structure in Sec. S1 of the supplementary material. First, in a synchronized rig the data depend on the trajectory only through alias sums, for any number of cameras, so adding synchronized cameras leaves the temporal bandwidth at $1/(2T)$. Second, in a K-group stagger, a trajectory with K alias components per set is identifiable if and only if every group triangulates, $H_p \succ 0$, *i.e.* has two cameras with non-parallel rays to the point. When every group triangulates, staggering thus raises the temporal bandwidth K-fold. With $N < 2K$, some groups have one camera and cannot triangulate.

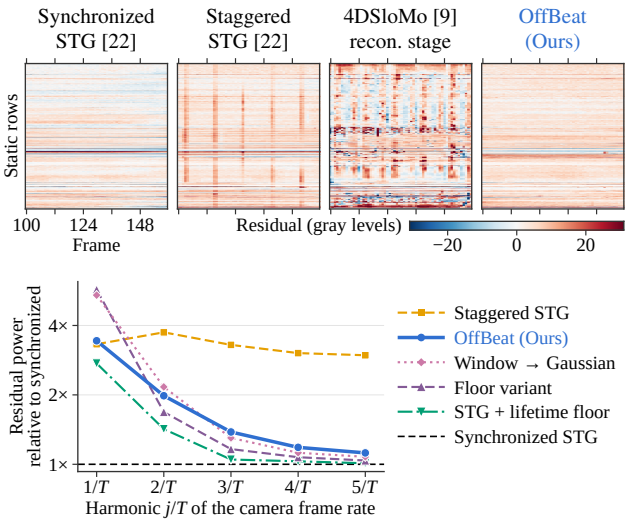


Figure 4. **Flicker on static pixels.** Top: test-view residual over test frames on Flame Salmon along the image column at 1/4 of the width. Ticks mark synchronized frames; vertical stripes show static pixels that change together. Bottom: static-pixel residual power at the harmonics j/T relative to synchronized STG, geometric mean over the six N3DV scenes. Staggered methods use the OffBeat schedule. Floor variant: STG with a lifetime floor, per-image timestamps and no opacity reset. Window → Gaussian: OffBeat with STG's Gaussian window and unscaled velocity.

A motion model over one camera frame interval pools all groups. Let a point's position over this period T be a degree-J polynomial in time d with coefficients $\theta \in \mathbb{R}^{3(J+1)}$, and let camera k see it at d_k. The Fisher information of θ is then $\sum_k \mathbf{b}(d_k)\mathbf{b}(d_k)^\top \otimes G_k$ with $\mathbf{b}(d) = (1, d, \ldots, d^J)$. For $J \geq 1$, it is singular in a synchronized rig and nonsingular once $J{+}1$ groups triangulate (Sec. S1). A polynomial per period replaces the K alias components by $J{+}1$ shared coefficients, as in trajectory triangulation [3, 29] and spatiotemporal bundle adjustment [42].

3.2. Trigger schedule

Position-ordered triggers. The trigger order leaves the identifiability condition of Sec. 3.1 unchanged but can undo the gain of staggering. Let cameras with centers $o_k = o_0 + s_k e$ on a line each fire once, at $t_k = t_0 + \alpha s_k$ with $\alpha \neq 0$, and let a point move as $X(t) = X_0 + v(t - t_0)$. For every $\lambda > 0$, the point with $X_0' = o_0 + \lambda(X_0 - o_0)$ and $v' = \lambda v + (1 - \lambda)e/\alpha$ has the same image in every camera. The Fisher information of (X_0, v) from one such sweep is singular for any number of cameras (proof in Sec. S3). The sweep thus confounds depth with motion, turning parallax into apparent motion (Fig. 2a). This is the ambiguity of a camera and a point both moving at constant velocity [3, 26, 29], and of a slowly moving camera [10]; in a rig, the schedule decides whether it arises. Repeated sweeps resolve it only for points keeping their velocity over several periods, unlike fast non-rigid motion. On N3DV, where the modular schedule sweeps both rows (Sec. S3), only their vertical offset keeps the information nonsingular.

Grouping. The partition into groups sets the information H_p of each instant, on which identifiability depends. With N3DV's 17 to 20 training cameras in 12 groups, at least four groups have a single camera under any partition, and the motion model pools them. Over the other groups, we maximize $\mathrm{mean}_p \lambda_p + \frac{1}{2} \min_p \lambda_p$, where λ_p is the working-volume average of the smallest eigenvalue of H_p (Sec. S3).

Ordering. For a point moving at constant velocity v over one period centered at μ and seen once per camera, pooling with $J{=}1$ gives the information of $(x,\ vT/2)$,

$$\mathcal{I}(\mu) = \sum_k \begin{bmatrix} 1 & d_k \\ d_k & d_k^2 \end{bmatrix} \otimes G_k, \qquad d_k = \frac{\mathrm{wrap}(\phi_k - \mu)}{T/2}, \tag{1}$$

where $d_k \in [-1, 1)$ is the wrapped trigger offset from μ in half periods. For novel views, we minimize $\max_{u,t} V + \frac{1}{2} \mathrm{mean}_{u,t} V$ with the predictive variance

$$V(u,t) = \mathbb{E}_{\mu,x}\, \mathrm{tr}\left(Q_u\, A\, \mathcal{I}(\mu)^{-1} A^\top\right), \quad A = [I,\ \tau I], \tag{2}$$

where τ is the wrapped offset of a trigger instant t from μ in half periods, Q_u is the information of a virtual view u midway between neighboring training cameras, and the expectation is over centers μ in one period and working-volume points x. Grouping and ordering follow the E- and G-optimality criteria of optimal design [14, 31], respectively. A local search optimizes them in seconds on a CPU (Alg. S1). Both need only the calibration; their weights $\frac{1}{2}$ hardly affect $\max V$ (Sec. S3). The modular schedule's $\max V$ per N3DV scene is 2.2–2.4 times ours.

3.3. Lifetime bound

Camera–time confound. Any error common to a group's views, such as a calibration or exposure error of a single-camera group, can be explained exactly by content present only when that group fires. If each group has its own error at a pixel, such as a color offset, the periodic signal h that equals the firing group's error reproduces it in every training image. In a synchronized rig, h is constant. A test camera belongs to no group and sees h as flicker at the camera frame rate and its harmonics (Fig. 4, Sec. 4.4). On N3DV, one or two cameras see each instant, so this explanation is always available.

Window condition. Primitive i with temporal center μ_i and window $w(t - \mu_i)$, such as STG's temporal

Table 1. **Comparison on N3DV** (K=12, 2.5 fps per camera, test frames 100–159). Dyn. PSNR (dynamic regions, Sec. 4.1) is also given over frames 100–148, where synchronized capture does not extrapolate. Best in bold, second best underlined.

Method	Dynamic-region PSNR↑ (dB)						Mean over scenes			
	Coffee	Cook	Beef	F. Salmon	F. Steak	Sear	Dyn. PSNR↑ 100–159	Dyn. PSNR↑ 100–148	PSNR↑	LPIPS↓
Synchronized capture										
STG [22]	23.17	25.88	25.95	19.98	23.84	27.16	24.33	24.97	30.57	.117
STG [22] + interpolation [21]	23.88	25.95	26.57	20.59	24.79	27.91	24.95	25.85	30.78	.129
Staggered capture, modular schedule										
4DSloMo [9], recon. stage	18.86	24.44	24.49	19.34	23.97	25.32	22.74	22.74	25.26	.218
4DSloMo [9], full pipeline	22.79	25.26	26.75	21.61	25.46	27.27	24.86	24.95	26.90	.186
Staggered capture, OffBeat schedule										
4DSloMo [9], recon. stage	21.20	26.49	26.36	21.63	27.08	28.20	25.16	25.14	27.11	.176
STG [22]	<u>24.94</u>	27.87	28.24	23.69	27.73	30.08	27.09	27.05	<u>30.85</u>	.116
STG [22], 2× iterations	24.65	<u>28.14</u>	<u>28.80</u>	<u>23.77</u>	<u>28.06</u>	<u>30.79</u>	<u>27.37</u>	<u>27.31</u>	30.79	<u>.108</u>
OffBeat (Ours)	**25.26**	**28.76**	**29.25**	**24.16**	**29.04**	**30.92**	**27.90**	**27.83**	**31.25**	**.104**

opacity [22], gives group p the total weight

$$\Omega_p(\mu_i) = \sum_m w\Big(\tfrac{pT}{K} + mT - \mu_i\Big) = (\Pi_T w)\Big(\tfrac{pT}{K} - \mu_i\Big), \tag{3}$$

where $\Pi_T w$ sums the copies of w shifted by whole periods (Fig. 2b). If $\Pi_T w$ varies, primitives can, for some K, weight the groups unequally and supply h. Prop. 1 (proof in Sec. S1) characterizes the windows that prevent this.

Proposition 1. *For a compactly supported window w, the following are equivalent: (a) $\Omega_p(\mu)$ is independent of p for every center μ and every K. (b) $\hat{w}(j/T) = 0$ for every integer $j \neq 0$, with $\hat{w}$ the Fourier transform of w. (c) For any centers μ_i and amplitudes b_i, $\Pi_T \sum_i b_i w(\cdot - \mu_i)$ is constant, so no set of primitives weights the groups unequally. (d) $w = \mathrm{box}_T * g$ for a compactly supported distribution g, with $\mathrm{box}_T = \mathbf{1}_{[-T/2,T/2)}$. No such window is shorter than T, and the shortest ones are shifted multiples of box_T.*

Prop. 1 thus bounds a primitive's lifetime below by T but leaves its motion sampled at K/T. Condition (b) is the partition-of-unity or Strang–Fix condition [38] at scale T, and (d) is the order-one case of the B-spline factorization [4]. The condition constrains the weights summed over all periods, over which a persistent single-camera error recurs, and leaves those within one period free.

3.4. Reconstruction on STG

We build on STG [22] and keep its loss and renderer. Each STG primitive i has a temporal center μ_i and scale a_i with temporal opacity $\exp(-(d/a_i)^2)$ at $d = t - \mu_i$, polynomial motion and rotation, and time-dependent features.

Lifetime floor. Gaussian windows violate Prop. 1(b). Wide ones have little power at the harmonics j/T (Sec. S1), but narrow ones can favor one group. Rendering with the temporal scale $\sqrt{a_i^2 + a_{\min}^2}$ in place of a_i, with $a_{\min} = T/4$, passes at most 29% of the power at the camera frame rate and under 1% at higher harmonics.

Period window. Since $\Pi_T \mathrm{box}_T = 1$, Prop. 1(d)⇒(c) holds for any integrable g, and a Gaussian g gives

$$w_i(t) = \frac{\operatorname{erf}\left((d + \frac{T}{2})/a_i\right) - \operatorname{erf}\left((d - \frac{T}{2})/a_i\right)}{2 \operatorname{erf}\left(T/(2a_i)\right)}. \tag{4}$$

For every center and scale, the period window meets the condition with no added parameter, gradients flow through the closed form, and it tends to the box window as $a_i \to 0$. We keep the floor on a_i because the window's gradient with respect to its center concentrates at the edges, and the floor spreads it from one to most of the 12 groups (Sec. S1).

Polynomial motion inside a window can reintroduce group-dependent components of h that Prop. 1(c) excludes for constant amplitudes, so we rescale STG's velocity to the displacement over half a window (Sec. S2).

Training. Two settings keep any update from depending on one group alone. With *per-image timestamps*, the timestamp is drawn per image, not per batch as in STG. With *no opacity reset*, the few cameras of one timestamp never have to re-estimate all opacities alone. We call STG with the floor and these two settings the *floor variant*.

Densification. Every training camera fires once in the period centered at t_0. For a small opaque primitive moving along $x + v(t - t_0)$ with a period window centered at t_0, let r_k be the residual at the pixel it covers in camera k when this camera fires. For a locally constant render, the primitive's best color lowers the squared error by $\Gamma(x, v, t_0) = \left|\sum_k r_k\right|^2/N$. This is the shift-and-stack statistic of track-before-detect [33] and, to first order in the opacity, the dual certificate of conditional gradient on measures [5, 6]. A residual r shared by m cameras scores $m^2|r|^2/N$, so errors common to one group (Sec. 3.3) rank low (Fig. 2c). Instead of STG's guided sampling along high-error rays, we add primitives at the maxima of Γ over depth and lateral velocity (Alg. S2), keeping STG's cloning and splitting. Other

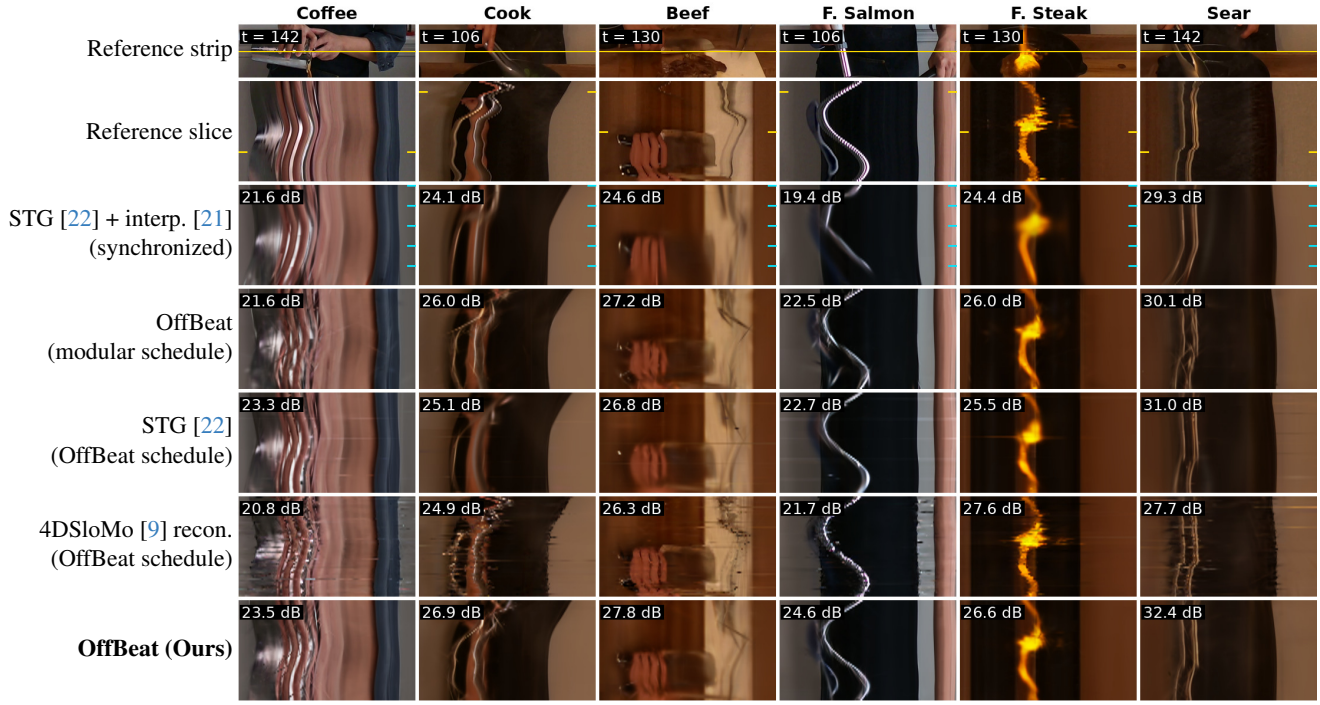


Figure 5. ***x–t* slices on N3DV** (Neural 3D Video). Top: strip at the yellow tick, midway between synchronized frames (cyan). Bottom: slices along its yellow line, time downward; slice PSNR in dB. Per scene, we pick the strip, instant and line of largest ground-truth change.

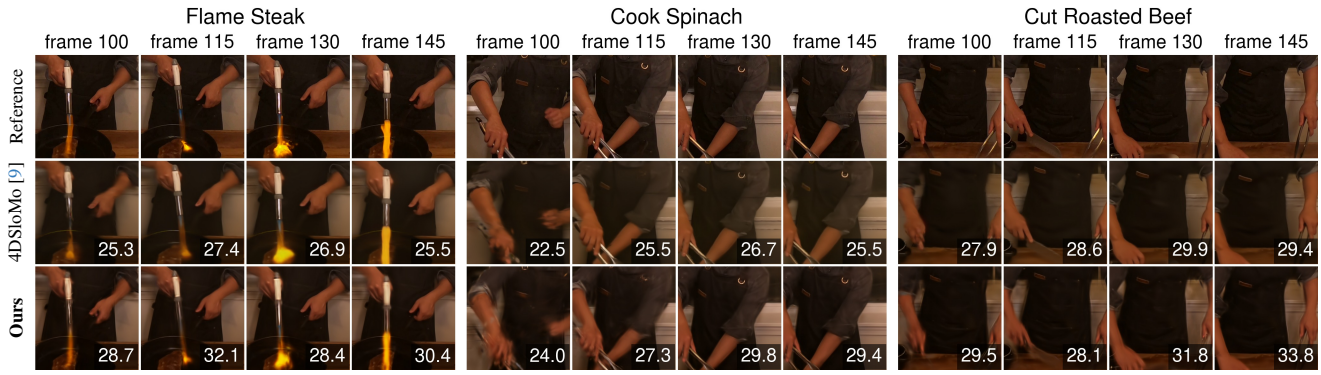


Figure 6. **OffBeat against the full 4DSloMo pipeline and schedule.** Crops around the x–t slice lines, every 15 frames; crop PSNR in dB. OffBeat leads 4DSloMo in Dyn. PSNR and PSNR on every test frame of all N3DV scenes, but not always in crop PSNR (Sec. S5).

methods densify by gradients or errors [13, 25, 35, 43], temporal visibility [36], or appearance change [34].

4. Experiments

4.1. Setup

Data and protocol. N3DV [20] contains six dynamic scenes. We hold out camera 0 for testing and train on the other 17–20 cameras. Following 4DSloMo [9], we subsample each 30 fps video to every 12th frame (2.5 fps), with a separate offset per camera for staggered capture and a common one for synchronized capture. We report test frames 100–159 unless stated otherwise and validate on frames 40–99 of Flame Steak and Cook Spinach. PanopticSports [12, 24] has 27 training and 4 test cameras, and we use six sequences with test frames 0–59 and $K{=}12$, validating on frames 90–149 of Softball and Basketball. Each method is trained once per scene, and Sec. S6 reports three seeds.

Compared methods. For synchronized capture, we train STG and *STG + interpolation*, STG trained on videos in which AMT [21] fills the 11 frames between synchronized frames. RetimeGS [44], for which we found no public code, reports this recipe with FILM [32] as its strongest N3DV baseline, 0.05 dB below its own result. For staggered capture, we compare OffBeat with STG, also trained with twice the iterations, and with 4DSloMo [9], using its

official code and settings. We run its reconstruction stage, a native 4D Gaussian model [49], on both schedules and on the synchronized frames of N3DV (where it is weaker than STG); we run its full pipeline with video diffusion on all six N3DV scenes (Sec. S5). All STG-based methods share the initialization, renderer and clock alignment (Sec. S8) and are trained for 10k iterations unless stated otherwise. OffBeat trains in 423–481 s per scene on one RTX 4090.

Metrics. At the test view, we report PSNR, AlexNet LPIPS [51] and dynamic-region PSNR (Dyn. PSNR, as in FreeTimeGS [45]); Sec. S6 reports SSIM. Dynamic pixels are those whose ground-truth luma has a temporal standard deviation above 0.05 over the 60 test frames, dilated by 5 px. Per scene and metric, the best synchronized method is the strongest of STG, STG + interpolation, the floor variant and, on N3DV, our reconstruction, all from synchronized capture. Sec. S6 gives per-scene results with paired bootstrap intervals [17]. Over 20 pairs of mask threshold and dilation, OffBeat's Dyn. PSNR difference keeps its sign against synchronized STG, STG + interpolation, STG on our schedule (except on PanopticSports) and the floor variant (negative on PanopticSports; Sec. S6).

4.2. N3DV

Staggered against synchronized capture. Between the first and last synchronized frame, OffBeat outperforms the best synchronized method by 1.98 dB in mean Dyn. PSNR (Table 1, Fig. 3) and on five of six scenes (Sec. S6). At the five synchronized frames, the better of STG and STG + interpolation in each scene is 1.23 dB above OffBeat in Dyn. PSNR but 2.20 dB below on the 44 frames in between (Sec. S6). In Fig. 5, interpolation flattens the traces, the modular schedule fragments them, and STG on our schedule adds single-frame streaks. 4DSloMo's reconstruction stage is sharp but jitters, and its slice PSNR on Flame Steak is higher than ours. OffBeat follows the trajectories.

Training length and camera frame rate. Synchronized STG trained twice as long remains 3.66 dB below OffBeat in Dyn. PSNR (Sec. S6). With the official 30k iterations for both, OffBeat outperforms STG + interpolation between the first and last synchronized frame by 2.29 dB in Dyn. PSNR on N3DV and by 0.99 dB on PanopticSports (Sec. S8). With twice the camera frame rate (5 fps), synchronized STG is still 1.02 dB below OffBeat at 2.5 fps between its first and last synchronized frame. At 10 fps it is 0.36 dB better there, and at 30 fps it is 0.60 dB better over all test frames, or 1.20 dB if it alone gets 30k iterations (Sec. S6).

Schedule. With our reconstruction, replacing our schedule by the modular one lowers Dyn. PSNR by 1.29 dB and PSNR by 0.46 dB. It also lowers the Dyn. PSNR of STG and of the 4DSloMo reconstruction stage by 1.02 and 2.42 dB. On N3DV, 48 stratified random schedules with our group sizes are 0.39 dB below ours on average, recovering 69% of the modular schedule's loss. Our two criteria rank them only weakly, and three of them outperform ours (Sec. S3).

Table 2. **Comparison on PanopticSports** (K=12, test camera 0, test frames 0–59, mean over six sequences). Best in bold, second best underlined. †Exception: per sequence and metric, the better of the modular and OffBeat schedules.

Method	Dyn. PSNR↑	PSNR↑	LPIPS↓
Synchronized capture			
STG [22]	22.31	26.87	.140
STG [22] + interpolation [21]	23.09	27.59	.159
Staggered capture, OffBeat schedule			
4DSloMo [9], recon. stage†	23.30	24.30	.310
STG [22]	<u>23.86</u>	<u>27.76</u>	<u>.138</u>
STG [22], 2× iterations	23.75	27.52	**.130**
OffBeat (Ours)	**23.98**	**27.98**	<u>.138</u>

4DSloMo. The full 4DSloMo pipeline, with its own schedule, is 3.04 dB below OffBeat in Dyn. PSNR and 4.35 dB in PSNR over all six scenes (Table 1, Fig. 6). Its diffusion stage (about a day per scene) raises the Dyn. PSNR of its own reconstruction stage by 2.03 dB; our run of that stage matches it within 0.09 dB on average. Sec. S5 relates 4DSloMo's published 33.48 dB to our protocol.

4.3. PanopticSports

On PanopticSports, the 27 training cameras form groups of two or three, each triangulating (Sec. 3.1). Between the first and last synchronized frame (frames 0–48, Sec. S6), OffBeat outperforms synchronized STG by 1.30 dB in Dyn. PSNR and is on par in Dyn. PSNR and PSNR with STG + interpolation. In this range, synchronized STG is slightly better in LPIPS, and STG + interpolation is better in full-frame and dynamic-region SSIM on every sequence. In Table 2, 4DSloMo's reconstruction stage is 0.68 dB lower in Dyn. PSNR and 3.69 dB in PSNR (Fig. 7), but better in Dyn. PSNR on Softball and Tennis, where OffBeat blurs the arms, bat and racket. At camera 0, the gain comes from staggering: unlike on N3DV, OffBeat is on par with STG on our schedule and its floor variant (Sec. S6). At the other three test cameras, OffBeat outperforms STG on our schedule on 13 of the 18 sequence–camera pairs in Dyn. PSNR. In frames 0–48, against the best synchronized method, it is on par or better in Dyn. PSNR and PSNR on 15 and 13 pairs but worse in LPIPS on 7 (Sec. S6).

4.4. Analysis

Ablation study. In Table 3, staggering raises the Dyn. PSNR of STG from 24.33 to 26.07 dB on the modular schedule and to 27.09 dB on ours. Most of the further gain comes from the lifetime floor (Prop. 1, 0.29 dB) and from

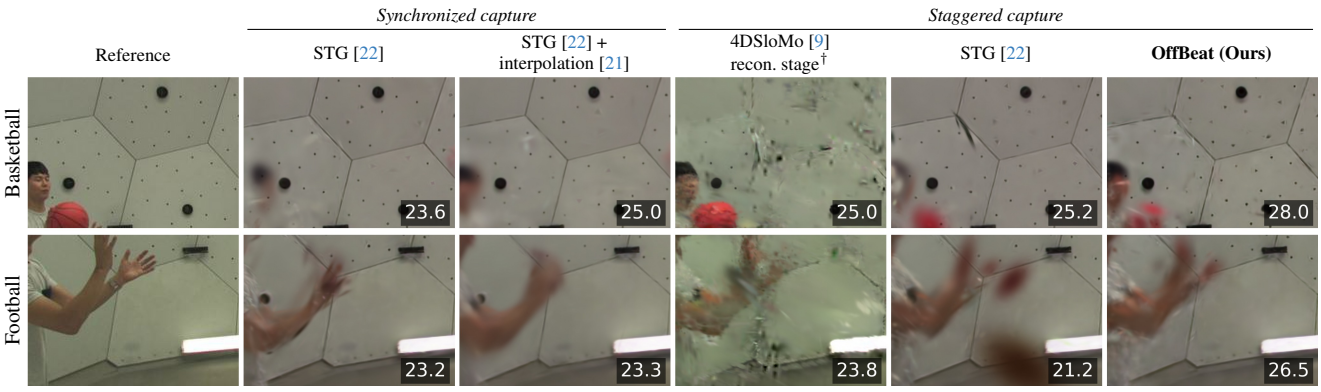


Figure 7. **PanopticSports, between synchronized frames.** Synchronized capture loses the ball (Basketball) and smears the hands (Football). 4DSloMo's reconstruction stage renders the ball sharply but clutters the wall, and STG on our schedule adds dark streaks and blobs. OffBeat has the highest crop PSNR (dB), but the ball and hands stay blurred. Crops, frames and the sequences shown are where OffBeat leads all other columns by the most, among the 10% of crops with the largest ground-truth change from the previous frame. †The better schedule in Dyn. PSNR. Sec. S6 shows Softball and Tennis, where 4DSloMo's reconstruction stage is better in Dyn. PSNR.

Table 3. **Ablation on N3DV** (test frames 100–159, mean over six scenes). *Per-image timestamps and no opacity reset (the floor variant). †With STG's unscaled velocity; its static-pixel residual power at $1/T$ is 1.58× OffBeat's (Fig. 4).

Variant	Dyn. PSNR↑	PSNR↑	LPIPS↓
Adding components			
STG [22], synchronized	24.33	30.57	.117
STG [22], modular schedule	26.07	30.47	.125
STG [22], OffBeat schedule	27.09	30.85	.116
+ lifetime floor	27.38	31.13	.115
+ training settings*	27.73	31.12	.108
+ window, densification (Ours)	27.90	31.25	.104
Replacing one component			
window → Gaussian†	27.91	31.30	.102
densification → STG's	27.63	31.03	.111
schedule → modular	26.61	30.79	.111
schedule → synchronized	23.80	30.06	.118

per-image timestamps with no opacity reset (0.34 dB). The period window with our densification adds 0.17 dB, which comes from Flame Steak and Cook Spinach, whose frames 40–99 we used for validation. On the other four scenes, OffBeat is 0.07 dB below the floor variant in Dyn. PSNR but still outperforms the best synchronized method by 2.66 dB and STG on our schedule by 0.66 dB (Sec. S6). Under the same floor, Gaussian windows are on par in Dyn. PSNR and better in PSNR (0.05 dB) and LPIPS, whereas STG's guided sampling in place of our densification lowers Dyn. PSNR by 0.27 dB. In Dyn. PSNR, no component helps synchronized capture: the floor variant and our reconstruction differ from STG by -0.17 and -0.54 dB (Sec. S6).

Flicker at the camera frame rate. On static pixels (Fig. 4, bottom), STG on our schedule has 3.0–3.7× the residual power of synchronized STG at the first five harmonics. The lifetime floor alone lowers this to 1.01–1.05× at harmonics 3–5 and to 2.7× at the fundamental, which the training settings raise to 5.7× while improving Dyn. PSNR. OffBeat is at 3.4× at the fundamental, within STG's range, but 1.1× at the fifth harmonic, and on Flame Salmon lacks STG's single-frame stripes (Fig. 4, top). Gaussian windows in place of the period window raise the fundamental power on all six scenes, to 1.58× OffBeat's (Sec. S7).

5. Conclusion

We presented OffBeat, a 4D capture method that models a staggered rig as a multicoset sampler to follow motion a synchronized rig misses. Under this model, a position-ordered row confounds depth with motion, the calibration alone yields a schedule, and temporal windows weighting all groups equally for every K last at least T, a bound our closed-form period window satisfies. On N3DV, OffBeat outperforms the full 4DSloMo pipeline in dynamic-region and full-frame PSNR on every test frame. Between the first and last synchronized frame, it follows motion better on average than a synchronized rig of equal camera frame rate, which is better at its own frames. On PanopticSports, in the same range, OffBeat is comparable in PSNR to synchronized capture with interpolation.

Limitations and Future Work. We simulate staggered capture by subsampling 30 fps video, and the schedule needs calibration before capture. On 4DSloMo's real staggered recording, we test only the lifetime floor, in its reconstruction stage, and find no reliable gain. In that stage, the period window as used on STG lowers dynamic-region PSNR on an N3DV validation scene (Sec. S5). OffBeat is more sensitive than synchronized STG to simulated trigger-time and photometric errors, which a real rig adds (Secs. S7 and S8). Estimating trigger times and gains during reconstruction could lower this sensitivity. Content shorter than T cannot be represented, and the fastest motion

between a group's firings stays blurred. OffBeat fails on Juggle at one PanopticSports camera (Sec. S6).

LLM Usage Statement

The authors chose the research problem and the core method and directed the use of Claude (Anthropic) through the Claude Code agent. Claude wrote and reviewed the experiment code, ran and monitored the experiments, computed the statistics, kept an append-only experiment log, and drafted the text and figures, including this statement. The simulated views of the velocity test in Sec. S4 were generated with code written by Claude. The authors approved each experiment plan before it was run, read every result, and edited the text. Gemini (Google) was used only to critique drafts. The authors are fully responsible for the content.

References

[1] Amit Agrawal, Mohit Gupta, Ashok Veeraraghavan, and Srinivasa G. Narasimhan. Optimal coded sampling for temporal super-resolution. In *CVPR*, 2010. 2

[2] Cenek Albl, Zuzana Kukelova, Andrew Fitzgibbon, Jan Heller, Matej Smid, and Tomas Pajdla. On the two-view geometry of unsynchronized cameras. In *CVPR*, 2017. 2

[3] Shai Avidan and Amnon Shashua. Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):348–357, 2000. 2, 4

[4] Thierry Blu, Philippe Thévenaz, and Michael Unser. MOMS: Maximal-order interpolation of minimal support. *IEEE Transactions on Image Processing*, 10(7):1069–1080, 2001. 5

[5] Nicholas Boyd, Geoffrey Schiebinger, and Benjamin Recht. The alternating descent conditional gradient method for sparse inverse problems. *SIAM Journal on Optimization*, 27 (2):616–639, 2017. 5

[6] Kristian Bredies and Hanna Katriina Pikkarainen. Inverse problems in spaces of measures. *ESAIM: Control, Optimisation and Calculus of Variations*, 19(1):190–218, 2013. 5

[7] Felix A. Breuer, Martin Blaimer, Robin M. Heidemann, Matthias F. Mueller, Mark A. Griswold, and Peter M. Jakob. Controlled aliasing in parallel imaging results in higher acceleration (CAIPIRINHA) for multi-slice imaging. *Magnetic Resonance in Medicine*, 53(3):684–691, 2005. 2

[8] Yaron Caspi and Michal Irani. Spatio-temporal alignment of sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1409–1424, 2002. 2

[9] Yutian Chen, Shi Guo, Tianshuo Yang, Lihe Ding, Xiuyuan Yu, Jinwei Gu, and Tianfan Xue. 4DSloMo: 4D reconstruction for high speed scene with asynchronous capture. In *SIGGRAPH Asia*, 2025. arXiv:2507.05163. 2, 3, 4, 5, 6, 7, 8

[10] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. In *NeurIPS*, 2022. 4

[11] Huayu Huang, Banglei Guan, Yang Shang, and Qifeng Yu. 3D trajectory reconstruction of moving points based on asynchronous cameras. *arXiv preprint arXiv:2506.00541*, 2025. 2

[12] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic Studio: A massively multiview system for social motion capture. In *ICCV*, 2015. 6

[13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 2, 6

[14] Jack Kiefer. General equivalence theory for optimum designs (approximate theory). *The Annals of Statistics*, 2(5):849–879, 1974. 4

[15] Seoha Kim, Jeongmin Bae, Youngsik Yun, Hahyun Lee, Gun Bang, and Youngjung Uh. Sync-NeRF: Generalizing dynamic NeRFs to unsynchronized videos. In *AAAI*, 2024. 2

[16] Seung-gyeom Kim, Areum Kim, Yongjae Yoo, and Sukmin Yun. Implicit 4D Gaussian splatting for fast motion with large inter-frame displacements. In *ICLR*, 2026. 2

[17] Hans R. Künsch. The jackknife and the bootstrap for general stationary observations. *The Annals of Statistics*, 17(3): 1217–1241, 1989. 7

[18] Junoh Lee, Changyeon Won, Hyunjun Jung, Inhwan Bae, and Hae-Gon Jeon. Fully explicit dynamic Gaussian splatting. In *NeurIPS*, 2024. 2

[19] Yonghan Lee, Tsung-Wei Huang, Shiv Gehlot, Jaehoon Choi, Guan-Ming Su, and Dinesh Manocha. SyncTrack4D: Cross-video motion alignment and video synchronization for multi-video 4D Gaussian splatting, 2025. arXiv:2512.04315. 2

[20] Tianye Li, Mira Slavcheva, Michael Zollhöfer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. Neural 3D video synthesis from multi-view video. In *CVPR*, 2022. 2, 6

[21] Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. AMT: All-pairs multi-field transforms for efficient frame interpolation. In *CVPR*, 2023. 1, 3, 5, 6, 7, 8

[22] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime Gaussian feature splatting for real-time dynamic view synthesis. In *CVPR*, 2024. 1, 2, 3, 4, 5, 6, 7, 8

[23] Zhanfeng Liao, Jiajun Zhang, Hanzhang Tu, Zhixi Wang, Yunqi Gao, Hongwen Zhang, and Yebin Liu. SharpTimeGS: Sharp and stable dynamic Gaussian splatting via lifespan modulation. In *CVPR*, 2026. 2

[24] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3D Gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024. 2, 6

[25] Yanzhe Lyu, Kai Cheng, Xin Kang, and Xuejin Chen. ResGS: Residual densification of 3D Gaussian for efficient detail recovery. In *ICCV*, 2025. 6

[26] Steven C. Nardone and Vincent J. Aidala. Observability criteria for bearings-only target motion analysis. *IEEE Transactions on Aerospace and Electronic Systems*, AES-17(2): 162–166, 1981. 4

[27] David Novikov, Eilon Vaknin, Narek Tumanyan, and Mark Sheinin. Color-encoded illumination for high-speed volumetric scene reconstruction. In *CVPR*, 2026. 2

[28] Athanasios Papoulis. Generalized sampling expansion. *IEEE Transactions on Circuits and Systems*, 24(11):652–654, 1977. 2, 3

[29] Hyun Soo Park, Takaaki Shiratori, Iain Matthews, and Yaser Sheikh. 3D reconstruction of a moving point from a series of 2D projections. In *ECCV*, 2010. 2, 4

[30] Klaas P. Pruessmann, Markus Weiger, Markus B. Scheidegger, and Peter Boesiger. SENSE: Sensitivity encoding for fast MRI. *Magnetic Resonance in Medicine*, 42(5):952–962, 1999. 2, 3

[31] Friedrich Pukelsheim. *Optimal Design of Experiments*. SIAM, 2006. Reprint of the 1993 Wiley edition. 4

[32] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. FILM: Frame interpolation for large motion. In *ECCV*, 2022. 6

[33] Irving S. Reed, Robert M. Gagliardi, and Huey M. Shao. Application of three-dimensional filtering to moving target detection. *IEEE Transactions on Aerospace and Electronic Systems*, 19(6):898–905, 1983. 5

[34] Minhyeok Roh, Minseo Lee, Jun Young Jeong, Byeonghyeon Lee, Lucas Yunkyu Lee, Jaesik Park, and Eunbyung Park. Dusting the phantom: Gaussian allocation for fading dynamic content. *arXiv preprint arXiv:2610.21302*, 2026. 6

[35] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. Revising densification in Gaussian splatting. In *ECCV*, 2024. 6

[36] Vikram Sandu, Mayurdeep Pathak, and Rajiv Soundararajan. Temporally aware densification for dynamic 3D Gaussian splatting. In *ECCV*, 2026. 6

[37] Eli Shechtman, Yaron Caspi, and Michal Irani. Space-time super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):531–545, 2005. 2

[38] Gilbert Strang and George Fix. A Fourier analysis of the finite element variational method. In *Constructive Aspects of Functional Analysis*, pages 793–840. Cremonese, 1973. 5

[39] Hang Su, Yunlong Feng, Daniel Gehrig, Panfeng Jiang, Ling Gao, Xavier Lagorce, and Laurent Kneip. A linear N-point solver for structure and motion from asynchronous tracks. In *ICCV*, 2025. 2

[40] Jeffrey Tsao, Peter Boesiger, and Klaas P. Pruessmann. k-t BLAST and k-t SENSE: Dynamic MRI with high frame rate exploiting spatiotemporal correlations. *Magnetic Resonance in Medicine*, 50(5):1031–1042, 2003. 2

[41] Raman Venkataramani and Yoram Bresler. Optimal sub-Nyquist nonuniform sampling and reconstruction for multiband signals. *IEEE Transactions on Signal Processing*, 49 (10):2301–2313, 2001. 2

[42] Minh Vo, Srinivasa G. Narasimhan, and Yaser Sheikh. Spatiotemporal bundle adjustment for dynamic 3D reconstruction. In *CVPR*, 2016. 2, 4

[43] Peihao Wang, Yuehao Wang, Dilin Wang, Sreyas Mohan, Zhiwen Fan, Lemeng Wu, Ruisi Cai, Yu-Ying Yeh, Zhangyang Wang, Qiang Liu, and Rakesh Ranjan. Steepest descent density control for compact 3D Gaussian splatting. In *CVPR*, 2025. 6

[44] Xuezhen Wang, Li Ma, Yulin Shen, Zeyu Wang, and Pedro V. Sander. RetimeGS: Continuous-time reconstruction of 4D Gaussian splatting. In *CVPR*, 2026. arXiv:2603.13783. 2, 6

[45] Yifan Wang, Peishan Yang, Zhen Xu, Jiaming Sun, Zhanhua Zhang, Yong Chen, Hujun Bao, Sida Peng, and Xiaowei Zhou. FreeTimeGS: Free Gaussian primitives at anytime anywhere for dynamic scene reconstruction. In *CVPR*, 2025. 2, 7

[46] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. High performance imaging using large camera arrays. *ACM Transactions on Graphics*, 24 (3):765–776, 2005. 2

[47] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4D Gaussian splatting for real-time dynamic scene rendering. In *CVPR*, 2024. 2

[48] Zhixin Xu, Hengyu Zhou, Yuan Liu, Wenhan Xue, Hao Pan, Wenping Wang, and Bin Wang. Dynamic Gaussian scene reconstruction from unsynchronized videos. In *AAAI*, 2026. arXiv:2511.11175. 2

[49] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4D Gaussian splatting. In *ICLR*, 2024. 2, 7

[50] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-Splatting: Alias-free 3D Gaussian splatting. In *CVPR*, 2024. 2

[51] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7

[52] Enliang Zheng, Dinghuang Ji, Enrique Dunn, and Jan-Michael Frahm. Sparse dynamic 3D reconstruction from unsynchronized videos. In *ICCV*, 2015. 2

[53] Junyu Zhu, Hao Zhu, Xinzhuo Zhang, Hongdong Li, Zhan Ma, and Xun Cao. ASTRA: Asynchronous spatio-temporal reconstruction via trajectory alignment. *arXiv preprint arXiv:2608.02006*, 2026. 2