Motion-from-Blur: 3D Shape and Motion Estimation of Motion-blurred Objects in Videos

Denys Rozumnyi[1,4] Martin R. Oswald[1,2] Vittorio Ferrari[3] Marc Pollefeys[1]
[1]Department of Computer Science, ETH Zurich [2]University of Amsterdam
[3]Google Research [4]Czech Technical University in Prague
{denys.rozumnyi,martin.oswald,marc.pollefeys}@inf.ethz.ch vittoferrari@google.com

Abstract

We propose a method for jointly estimating the 3D motion, 3D shape, and appearance of highly motion-blurred objects from a video. To this end, we model the blurred appearance of a fast moving object in a generative fashion by parametrizing its 3D position, rotation, velocity, acceleration, bounces, shape, and texture over the duration of a predefined time window spanning multiple frames. Using differentiable rendering, we are able to estimate all parameters by minimizing the pixel-wise reprojection error to the input video via backpropagating through a rendering pipeline that accounts for motion blur by averaging the graphics output over short time intervals. For that purpose, we also estimate the camera exposure gap time within the same optimization. To account for abrupt motion changes like bounces, we model the motion trajectory as a piece-wise polynomial, and we are able to estimate the specific time of the bounce at sub-frame accuracy. Experiments on established benchmark datasets demonstrate that our method outperforms previous methods for fast moving object deblurring and 3D reconstruction.

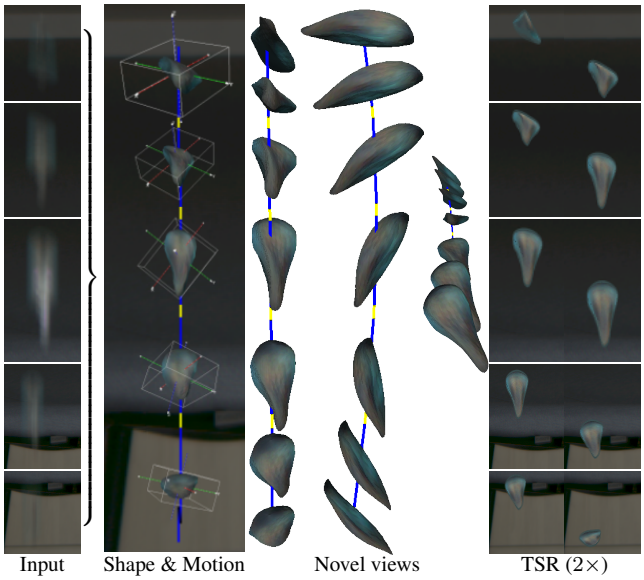

Figure 1. **Reconstructing 3D shape and motion of a motion-blurred falling key.** We jointly optimize over multiple input frames to estimate a single 3D textured mesh and corresponding motion model (*blue*: observed trajectory, *yellow*: the exposure gap). Temporal super-resolution (TSR) is one of the applications of the proposed Motion-from-Blur method.

1. Introduction

3D object reconstruction from 2D images is one of the key tasks in computer vision [20, 32, 33, 36]. It allows better modeling of the underlying 3D world. Applications of 3D object reconstruction are broad, ranging from robotic mapping [7] to augmented reality [42]. Even though some recent methods deal with the extreme and under-constrained case of reconstructing 3D objects from a single 2D image [39, 43], most methods take advantage of a multi-view setting [20, 32, 33, 36]. However, all generic 3D object reconstruction methods assume that the object moves slowly compared to the camera frame rate, resulting in sharp 2D images. The task of 3D object reconstruction becomes much more challenging when the object moves fast during the camera exposure time, resulting in a motion-blurred 2D image. The Shape-from-Blur (SfB) method [31] tackled this challenging scenario to extract 3D shape and motion from a single motion-blurred image of the object. This scenario is difficult because motion blur makes the input image noisier, and many high-frequency details are lost. On the other hand, even a single image gives potentially several views of the object, which are averaged by motion blur into one frame. SfB [31] explicitly modeled this phenomenon and successfully exploited it.

In this paper, we go beyond previous methods by estimating the 3D object's shape and its motion from a series of motion-blurred video frames. To achieve this, we optimize

all parameters jointly over multiple input frames (*i.e.* the object's 3D shape and texture, as well as its 3D motion). We tie up the object's 3D shape and texture to be constant over all frames. Due to the longer time intervals involved, we must model more complex object motions (3D translation and 3D rotation) than necessary for a single motion-blurred frame [31], *e.g.* the acceleration of a falling object (Fig. 1), or a ball bouncing against a wall (Fig. 3). Using multiple frames also comes with an additional challenge: the camera shutter opens and closes in set time intervals, leading to a gap in the object's visible trajectory and appearance. To properly succeed in our task, we must also recover this exposure gap. For a single frame only (as in [31]), the motion direction (forward vs. backward motion along the estimated axis) is ambiguous. For instance, in Fig. 1, the key could be translating from top to bottom or vice-versa, both resulting in the same input image. Since we consider multiple frames jointly, the motion direction is no longer ambiguous and can always be recovered. Moreover, for rotating objects, we can reconstruct a more complete 3D model as we can integrate more observations covering its total surface. In contrast, previous single-frame work [31] produces strong artifacts on unseen parts. An example of our method's output and an application to temporal super-resolution is shown in Fig. 1. To summarize, we make the following contributions:

(1) We propose a method called Motion-from-Blur (MfB) that jointly estimates the 3D motion, 3D shape, and texture of motion-blurred objects in videos by optimizing over multiple blurred frames. Motion-from-Blur is the first method to optimize over a video sequence instead of a single frame. The source code is available at github.com/rozumden/MotionFromBlur.

(2) Our multi-frame optimization enables the estimation of the motion direction as well as more complex object motions such as acceleration and abrupt direction changes, *e.g.* bounces, for both 3D translation and 3D rotation. Moreover, compared to single-frame approaches, our estimates are also more consistent over time, with always correct motion direction, and more complete 3D shape reconstruction.

(3) As a requirement to model multiple frames, we estimate the exposure gap as part of the proposed optimization.

2. Related work

Many methods have been proposed for generic deblurring, *e.g.* [3, 11, 17–19, 22, 38, 44]. A related task of frame interpolation or temporal super-resolution is studied in [4, 6, 9, 10, 21, 22, 34, 35]. However, none of the generic deblurring methods work on extremely motion-blurred objects as shown in [27], and specific methods are required.

We focus on deblurring and 3D reconstruction of highly motion-blurred objects. These are called fast moving objects as defined in [27] – objects that move over distances larger than their size within the exposure time of one image. Detection and tracking of such objects are usually done by classical image processing methods [14, 26, 27] or more recently by deep learning [29, 45]. A model-based approach to high-speed tracking with a specialized time-of-flight camera is studied in [37].

Single-frame deblurring of fast moving objects. The first methods for fast moving object deblurring [15, 27] assumed an object with a constant 2D appearance F and 2D shape mask M. Hence, the object was represented by a single 2D image patch that could only be rigidly translated and rotated in 2D. They defined the image formation model for such objects as the blending of the blurred object appearance F and the background B:

$$I = H * F + (1 - H * M) \cdot B \ , \tag{1}$$

where the motion blur is modeled by the convolution of the sharp object appearance F and its trajectory, defined by the blur kernel H. Several follow-up methods [13, 14, 25, 26, 28, 29, 40] were proposed to solve for (F, M, H) given the input image I and background B. They approximate the solution in a least-squares sense by energy minimization with suitable regularizers summarized by function $\mathrm{reg}(\cdot)$:

$$\min_{F,M,H} \frac{1}{2} \| H * F + (1 - H * M) \cdot B - I \|_2^2 + \mathrm{reg}(F, M, H) \ . \tag{2}$$

As common in blind deblurring problems [15], they deploy alternating minimization w.r.t. object (F, M) and trajectory H separately in a loop. Optimization is made possible thanks to many regularizers such as appearance total variation, blur kernel sparsity [14, 15, 26], motion blur prior for curves [40], appearance and mask rotational symmetry [28], among others. All of these methods share the same drawback that stems from the underlying image formation model (1), which assumes a constant 2D object appearance.

TbD-3D [28] extended the image formation model to support fast moving objects with a piece-wise constant 2D appearance as

$$I = \sum_{\tau} H_\tau * F_\tau + \left(1 - \sum_{\tau} H_\tau * M_\tau\right) \cdot B \ , \tag{3}$$

where the trajectory is split into several pieces H_τ, assuming that along each piece the object appearance F_τ and mask M_τ are constant. All unknowns are again estimated by energy minimization with additional problem-specific priors, *e.g.* object appearance in neighboring pieces is similar.

Later, DeFMO [30] was the first learning-based method for fast moving object deblurring, and it generalized the image formation model further to objects with a 2D appearance that can change arbitrarily along the trajectory:

$$I = \int_0^1 F_\tau \cdot M_\tau \, \mathrm{d}\tau + \left(1 - \int_0^1 M_\tau \, \mathrm{d}\tau\right) \cdot B \ , \tag{4}$$

where object appearance F_τ and mask M_τ are modeled by an encoder-decoder network. The network places (F_τ, M_τ) at the right image location, directly encoding the object trajectory. Although trained on synthetic ShapeNet data [1], DeFMO was shown to generalize to real-world images.

Single-frame 3D reconstruction of fast moving objects. The only prior work capable of 3D reconstruction of fast moving objects is Shape-from-Blur [31]. Instead of merely recovering the 2D object projections (F_τ, M_τ), they reconstruct the object's 3D shape mesh Θ as well as 3D motion. The latter is represented as the 3D translation $\mathbf{t}$ and 3D rotation $\mathbf{r}$, defining the object's pose at the beginning of the exposure time ($\tau = 0$), and the offsets $\Delta\mathbf{t}$ and $\Delta\mathbf{r}$, moving the object to its pose at the end of the exposure time ($\tau = 1$). With these definitions, the image formation model becomes

$$I = \int_0^1 \mathcal{R}_F\big(\mathcal{M}(\Theta, \mathbf{r} + \tau \cdot \Delta\mathbf{r}, \mathbf{t} + \tau \cdot \Delta\mathbf{t})\big)\, \mathrm{d}\tau + \\ + \Big(1 - \int_0^1 \mathcal{R}_S\big(\mathcal{M}(\Theta, \mathbf{r} + \tau \cdot \Delta\mathbf{r}, \mathbf{t} + \tau \cdot \Delta\mathbf{t})\big)\, \mathrm{d}\tau\Big) \cdot B \ , \tag{5}$$

where the function $\mathcal{M}$ transforms the mesh Θ by the given 3D translation and 3D rotation. Energy minimization is constructed from (5) to find the mesh and motion parameters that would re-render the input image I as closely as possible. To make minimization feasible, mesh rendering is made differentiable using Differentialbe Interpolation-Based Rendering [2], denoted by $\mathcal{R}_F$ and $\mathcal{R}_S$ for the appearance and 2D object silhouette, respectively. To differentiate from 2D masks M_τ, silhouettes denote real renderings of a 3D object mesh. In contrast to Shape-from-Blur, our method models more complex trajectories, estimates the exposure gap, and takes into account several frames jointly, thereby allowing temporally consistent predictions and more completely reconstructed 3D shape models.

3D shape from sharp images. Many methods for 3D reconstruction have been proposed, both for single-frame [5, 24, 39, 41, 43] and multi-frame setting [20, 32, 33, 36]. But these methods assume sharp objects in the scene (the methods listed in previous paragraphs are the only ones dedicated to fast moving objects). In other words, they either assume that an object moves slowly compared to the camera frame rate (or, equivalently, that the camera moves slowly).

3. Method

When images are captured by a conventional camera, the camera opens its shutter to allow the right amount of light to reach the camera sensor. Then, the shutter closes, and the whole process is repeated until the required number of frames is captured. This physical reality of the camera capturing process leads to two phenomena, which we model and exploit in our optimization. The first one is the motion blur that appears when the object moves while the shutter is open. The second one is the exposure gap that makes the camera 'blind' when the shutter is closed, thus not observing the moving object for some parts of its motion.

We assume the input is a video stream $V = \{I_1, \ldots, I_N\}$ of N RGB images depicting a fast moving object. The desired output of our method is a single textured 3D object mesh Θ, its motion parameters Ω consisting of a continuous 3D translation and 3D rotation at every point in time τ during the video duration, and the exposure gap ϵ (a real-valued parameter). Sec. 3.1 introduces these parameters and a video formation model to generate video frames for given parameters. In case we know the real values of all parameters, we could render the input video V. Then, in Sec. 3.2, we show how to optimize these parameters to re-render the input video frames as closely as possible.

3.1. Modeling

Mesh modeling. The mesh parameters Θ consist of an index to a prototype mesh, vertex offsets from its initial vertex positions to deform the mesh, and the texture map. We use a set of prototype meshes to account for varying mesh complexity and different genus numbers. Our set of prototype meshes contains a torus and two spheres with a different number of vertices. The texture mapping from vertices to the 2D location on the texture map is assumed to be fixed. Similarly, the mesh triangular faces consist of fixed sets of edges that connect vertices.

Motion modeling. The object motion Ω is composed of continuous 3D translations $\mathcal{T}(\tau) \in \mathbb{R}^3$ and 3D rotations represented by quaternions $\mathcal{Q}(\tau) \in \mathbb{R}^4$. Both translations and rotations are viewed from the camera perspective, which is assumed to be static. We assume that they are defined at all points in time $\tau \in [0, 1]$, spanning the duration of the entire input video. We implement the functions $\mathcal{T}(\tau)$ and $\mathcal{Q}(\tau)$ as piece-wise polynomials, and their parameters are the polynomial coefficients. More precisely, we use piece-wise quadratic functions with two connected pieces, which are able to model one bounce, as well as accelerating motions (*e.g.* a falling object).

Exposure modeling. We denote the *exposure gap* as a real-valued parameter $\epsilon \in [0, 1]$ that represents the fraction of the duration of a frame during which the camera shutter is closed. In other words, it is the duration of the closed shutter divided by the duration of one shutter cycle. A hypothetical full exposure camera that never closes its shutter would result in $\epsilon = 0$. In most cases, conventional cameras would set their exposure gap ϵ close to 0 for dark environments to get as much light as possible and close to 1 for very bright environments to avoid overexposure. Typically, smaller exposure gaps ϵ lead to more motion blur.

Video formation model. The video formation model is

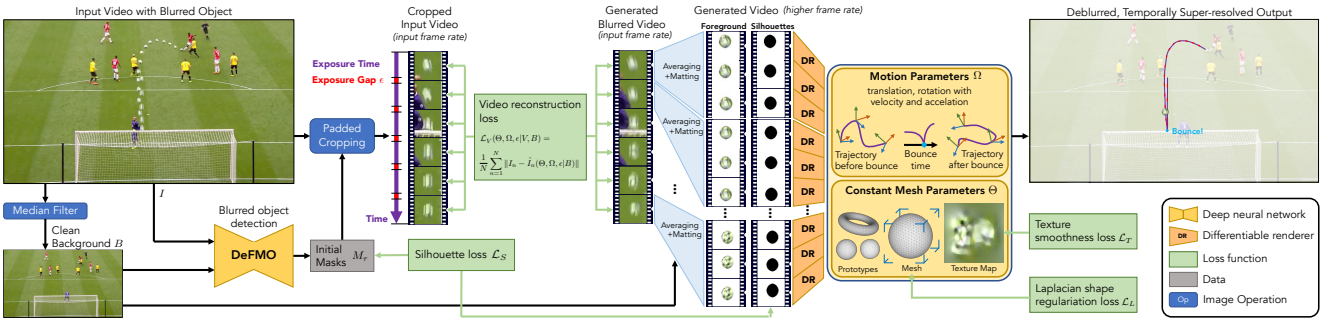

Figure 2. **Overview of Motion-from-Blur (MfB).** For a video of a motion-blurred object, we estimate its 3D motion, 3D shape, and texture. From *right to left*, the pipeline can be interpreted as a generative model: Starting from all parameters for an object and its motion, we render high-frame-rate videos with the object appearance (foreground) and its silhouette. Together with the known background, we generate a motion-blurred video of the object that should match the input video as good as possible. At test time, we optimize all object parameters (and the exposure gap) of this inverse problem by backpropagating the image differences through the differentiable renderer (*left to right*). We initialize the optimization using the DeFMO method [30], which provides rough silhouettes of the blurred object. MfB models a piece-wise smooth motion path to allow for a motion discontinuity like a bounce. Video source: YouTube.

the core of our method. It renders a video frame $\hat{I}_n$ for a given set of all above-mentioned parameters:

$$\hat{I}_n(\Theta,\Omega,\epsilon|B) = \int_{\frac{n-1}{N}}^{\frac{n-\epsilon}{N}} \mathcal{R}_F\Big(\mathcal{M}\Big(\Theta,\mathcal{Q}(\tau),\mathcal{T}(\tau)\Big)\Big)\,\mathrm{d}\tau + \\ + \left(1 - \int_{\frac{n-1}{N}}^{\frac{n-\epsilon}{N}} \mathcal{R}_S\Big(\mathcal{M}\Big(\Theta,\mathcal{Q}(\tau),\mathcal{T}(\tau)\Big)\Big)\,\mathrm{d}\tau\right)\cdot B\ , \tag{6}$$

where the interval bounds for frame $\hat{I}_n$ go from the beginning of its exposure time when the shutter opens at time $\tau = \frac{n-1}{N}$ to the end of its exposure time when the shutter closes at time $\tau = \frac{n-\epsilon}{N}$. Consequently, the object is not observed between $\tau = \frac{n-\epsilon}{N}$ and $\tau = \frac{n}{N}$. As defined previously, the function $\mathcal{M}$ first rotates the mesh Θ by the 3D rotation $\mathcal{Q}(\tau)$ and then moves it by the 3D translation $\mathcal{T}(\tau)$. Mesh rendering is implemented by Differentiable Interpolation-Based Rendering [2], denoted by $\mathcal{R}_F$ for the appearance and by $\mathcal{R}_S$ for the silhouette. Like all previous methods for fast moving object deblurring, we compute the background B as the median of all frames I_n in the input video V. Note that our modeling is a strict generalization of SfB [31] for the case of $N = 1$ and linear motion.

3.2. Model fitting

This section presents an optimization method to fit the introduced model to the given input video.

Loss function. The main driving force of the proposed approach is the video reconstruction loss

$$\mathcal{L}_V(\Theta,\Omega,\epsilon|V,B) = \frac{1}{N}\sum_{n=1}^{N} \|I_n - \hat{I}_n(\Theta,\Omega,\epsilon|B)\|_1\ . \tag{7}$$

This loss is low if the frames $\hat{I}_n$ rendered by our model via Eq. (6) closely look like the input frames I_n.

In order to make the optimization easier and well-behaved, we apply auxiliary loss terms and regularizers, similar to [31]. We briefly summarize them here and refer to [31] for details. The silhouette consistency loss $\mathcal{L}_S$ helps localize the object in the image faster and serves as initialization for estimating the 3D mesh and its translation. First, we run DeFMO [30] and use their estimated masks M_τ for approximate object location. To synchronize the motion direction (forward vs. backward) for DeFMO masks across frames, we minimize the distance between consecutive masks in adjacent frames. Then, $\mathcal{L}_S$ is defined as an intersection over union (IoU) between the DeFMO masks and 2D mesh silhouettes, rendered by our method:

$$\mathcal{L}_S = 1 - \int_0^1 \mathrm{IoU}\Big(M_\tau, \mathcal{R}_S\Big(\mathcal{M}\Big(\Theta,\mathcal{Q}(\tau),\mathcal{T}(\tau)\Big)\Big)\Big)\,\mathrm{d}\tau. \tag{8}$$

Furthermore, we add the commonly employed [2, 13, 26, 31, 41] total variation and Laplacian regularizers. Total variation $\mathcal{L}_T(\Theta)$ on texture maps encourages the model to produce smooth textures, and the Laplacian regularizer $\mathcal{L}_L(\Theta)$ promotes smooth meshes. Finally, the joint loss is a weighted sum of all four loss terms:

$$\mathcal{L}(\Theta,\Omega,\epsilon|V,B) = \lambda_V\cdot\mathcal{L}_V(\Theta,\Omega,\epsilon|V,B) + \mathcal{L}_T(\Theta) + \\ + \mathcal{L}_S(\Theta,\Omega,\epsilon|V,B) + \lambda_L\cdot\mathcal{L}_L(\Theta)\ . \tag{9}$$

Optimization. Fig. 2 shows an overview of the pipeline. We backpropagate the joint loss up to the mesh Θ, motion parameters Ω, and exposure gap ϵ. Optimization is done with ADAM [12] using a learning rate of 0.1. In the beginning, we run pre-optimization for at most 100 iterations with $\lambda_V = 0$, thus omitting the video reconstruction loss

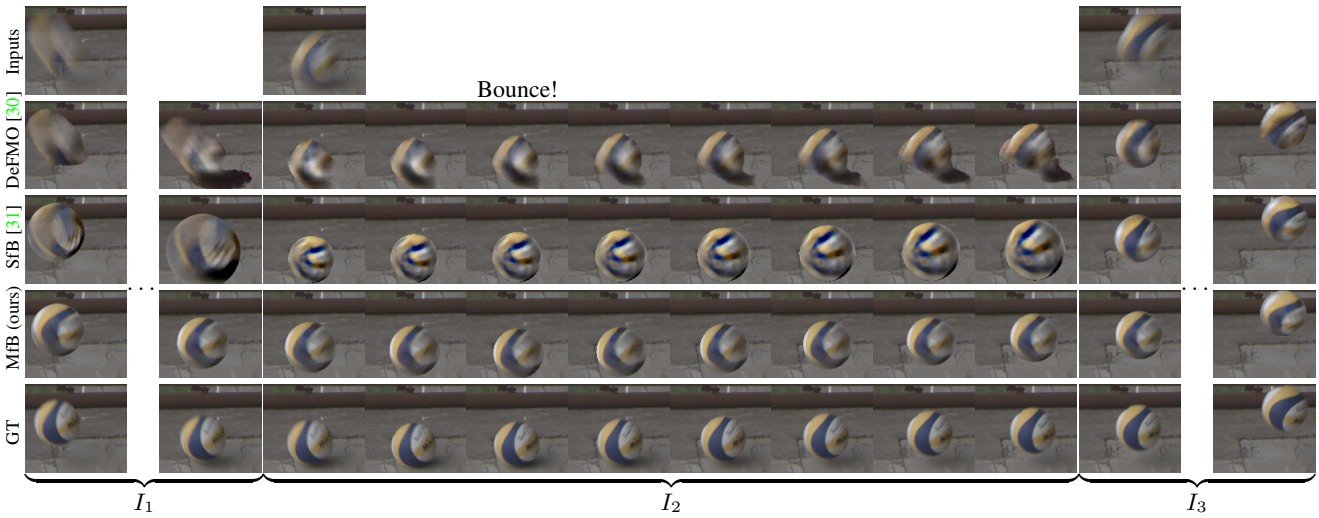

Figure 3. **Estimating 3D shape and motion of a motion-blurred volleyball, shown as temporal super-resolution.** The proposed Motion-from-Blur (MfB) method is the first to use multiple video frames during optimization and the first to model complex trajectories with bounces, accounting for the exposure gap. The previous methods for FMO deblurring (DeFMO) and single-frame 3D reconstruction (SfB) have difficulties reconstructing the bounce as they get confused by the ball's shadow due to the lack of multi-frame optimization.

and texture map updates. Pre-optimization stops when the silhouette loss $\mathcal{L}_S$ becomes < 0.3, meaning that the mesh silhouettes have average IoU > 0.7 with the DeFMO masks. This pre-optimization phase is required since the 3D translation has to put the mesh at approximately the right location in the image to get a training signal for the video reconstruction loss $\mathcal{L}_V$ to estimate the texture map, 3D object rotation, and 3D shape. The more video frames N are used, the more important this step becomes because the object's 2D location varies more across the frames. Experimentally, for $N > 2$ the optimization never converges without pre-optimization. We optimize over the mesh prototypes by running the optimization for each prototype and choosing the best one based on the lowest value of the video reconstruction loss (7). During optimization, the mesh is always kept in canonical space by normalizing the vertices to zero mean and unit variance. The main optimization is run for 1000 iterations using the full loss (9) with $\lambda_V = 1$. The hyperparameter λ_L of the Laplacian regularizer $\mathcal{L}_L$ is set to 1000 experimentally. Both the texture total variation $\mathcal{L}_T$ and silhouette consistency $\mathcal{L}_S$ losses have no weights since the default value of 1 worked well in our experiments.

Initialization. The mesh parameters Θ are initialized to the prototype shape with zero vertex offsets and a white texture map. The motion parameters Ω are initialized such that the object is placed in the middle of the image with zero rotation. Finally, the exposure gap ϵ is initialized to 0.1.

Implementation. We use PyTorch [23] with Kaolin [8] for differentiable rendering. All integrals in each frame are discretized by splitting time intervals into 8 evenly-spaced pieces. All experiments are run on an Nvidia GTX 1080Ti GPU with 60 seconds average runtime per frame.

4. Experiments

We evaluate our method's accuracy by measuring the deblurring quality on 3 real-world datasets from the fast moving object deblurring benchmark [30]. Since there are no real image datasets of fast moving objects with associated ground-truth 3D shapes and motion, we follow the protocol of [31] and evaluate the quality of reconstructed 3D meshes, 3D translations, and 3D rotations on a synthetic dataset.

Fast moving object deblurring benchmark. It consists of 3 datasets of varying difficulty. The easiest one is TbD [14] that contains mostly spherical objects with uniform color (12 sequences, total 471 frames). A more difficult dataset is TbD-3D [28] that contains mostly spherical objects with complex textures that move with significant 3D rotation (10 sequences, total 516 frames). The most difficult dataset is Falling Objects [13] with objects of various shapes and complex textures (6 sequences, total 94 frames). The ground truth for these datasets was recorded by a high-speed camera capturing the moving object without motion blur. Therefore, we have 8 high-speed frames for each frame input to our method. We measure the deblurring quality by reconstructing the high-speed camera footage as temporal super-resolution. For that, we apply the video formation model (6) at a 8 times finer temporal resolution by using the estimated object parameters after optimization on the input slow-speed frames. Then, the reconstructed high-speed camera frames and the ground-truth ones are compared by the Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) metrics. Additionally, these datasets contain ground-truth 2D object trajectories and 2D object masks. Therefore, we also measure the trajectory intersection over

Method	Falling Objects [13]			TbD-3D Dataset [28]			TbD Dataset [14]		
	TIoU↑	PSNR↑	SSIM↑	TIoU↑	PSNR↑	SSIM↑	TIoU↑	PSNR↑	SSIM↑
Jin *et al*. [10]	N / A	23.54	0.575	N / A	24.52	0.590	N / A	24.90	0.530
DeblurGAN-v2 [18]	N / A	23.36	0.588	N / A	23.58	0.603	N / A	24.27	0.537
TbD [14]	0.539	20.53	0.591	0.598	18.84	0.504	0.542	23.22	0.605
TbD-3D [28]	0.539	23.42	0.671	0.598	23.13	0.651	0.542	25.21	0.674
DeFMO [30]	0.684	26.83	0.753	0.879	26.23	0.699	0.550	25.57	0.602
SfB [31]	0.701	27.18	0.760	0.921	26.54	0.722	0.610	25.66	0.659
MfB (ours)	**0.772**	**27.54**	**0.765**	**0.927**	**26.57**	**0.728**	**0.614**	**26.63**	**0.678**

Table 1. **Fast moving object deblurring benchmark.** We compare the proposed MfB method to generic deblurring methods [10, 18] (no trajectory output, thus TIoU is undefined) and to methods specifically designed for fast moving object deblurring [14, 28, 30, 31].

		TIoU↑	PSNR↑	SSIM↑
full	SfB [31]	0.921	26.54	0.722
	MfB (ours)	**0.927**	**26.57**	**0.728**
bnc ±1	SfB [31]	0.892	21.77	0.628
	MfB (ours)	**0.902**	**25.01**	**0.643**
bnc ±0	SfB [31]	0.863	20.77	0.595
	MfB (ours)	**0.889**	**24.57**	**0.620**

Table 2. **Deblurring quality at bounces.** We compare scores on the full TbD-3D dataset [28], on a subset of frames at bounces (bnc ±0), and additionally on frames that are immediately before and after the bounce (bnc ±1). The proposed multi-frame MfB is significantly more accurate at bounces (Fig. 3) than the single-frame SfB, especially on the deblurring metric PSNR.

		$e_{\mathbf{t}}\downarrow$	$e_{\mathbf{r}}\downarrow$	$e_{\Theta}\downarrow$
$\sphericalangle$ 90°	SfB [31]	37.8 %	10.9°	3.0 %
	MfB (ours)	**20.0 %**	**6.4°**	**2.7 %**
$\sphericalangle$ 30°	SfB [31]	12.8 %	4.8°	2.3 %
	MfB (ours)	**8.8 %**	**3.7°**	**2.2 %**

Table 3. **Evaluating 3D translation, 3D rotation, and 3D shape on a synthetic dataset.** *First block:* dataset with at most 90° rotation over 3 frames, *second block:* at most 30° rotation. The error rate of MfB is half of the single-frame SfB on the large rotation dataset when measuring 3D translation $e_{\mathbf{t}}$ and 3D rotation $e_{\mathbf{r}}$ errors. MfB is still significantly better on the small rotation dataset.

union (TIoU), defined as the IoU between the ground-truth mask placed at the ground-truth 2D location and the reconstructed 2D location (averaged over time). We reconstruct the 2D object location for our method as the center of mass of the projected mesh silhouette at each high-speed frame.

We compare to various state-of-the-art methods: a generic deblurring method DeblurGAN-v2 [18], a generic method for temporal super-resolution [10], and methods designed for fast moving object deblurring [14, 28, 30, 31]. All compared methods use each video frame independently, whereas MfB is the first method to exploit multiple frames simultaneously. We run MfB in a temporal sliding window approach with $N = 3$ if not mentioned otherwise. For each frame, we always choose the window for which the video reconstruction loss (7) is the lowest, measured only on this frame (similar to the best prototype selection). This temporal sliding window approach reduces memory requirements and increases robustness to slightly moving camera and non-static background.

Table 1 presents the results. MfB outperforms all other methods on all three datasets and for all three metrics. Qualitatively, the estimated temporal super-resolution is more consistent compared to single-frame approaches since MfB explains all frames by a single 3D object mesh and texture (Fig. 5, $N = 7$). Novel view synthesis is also considerably better as the object outline is accurate from all viewpoints, and even sharp 90° angles of the box (Fig. 5, novel views) are clear. Interestingly, the previous state-of-the-art single-frame 3D reconstruction approach [31] produces several artifacts, inconsistencies, and produces an entirely incorrect 3D shape for object parts that are not visible in a single input frame. Moreover, DeFMO [30] and SfB [31] fail in the presence of shadows and specularities, whereas MfB better reconstructs the object due to additional constraints from neighboring frames (Fig. 5, I_1 and I_2).

Exposure gap consistency. We evaluate the variance of the exposure gap estimation over the sequence duration, averaged over all sequences. The value is very low ($\sigma^2 = 0.002$), indicating good consistency. Besides, the estimated exposure gap varies widely depending on the camera settings: $\tau = 0.05$ on the Falling Objects dataset, and $\tau = 0.7$ on the YouTube sequence in Fig. 4 (bottom). This highlights the need for modeling the exposure gap.

Evaluating at bounces. A unique new feature of our approach is its ability to model bounces, which results in better deblurring in those cases. Here, we evaluate this effect explicitly. To this end, we manually annotate the frames in which a bounce happens in the TbD-3D dataset [28] (the only dataset with relatively frequent bounces). Overall, we found 38 bounces from 516 frames in total from 10 sequences, which amounts to 7.4% chance of a bounce. Since the frames immediately before and after a bounce are usu-

ally affected too (*e.g.* due to a shadow as in Fig. 3), we also evaluate them, yielding a total of 114 frames (22%). As shown in Table 2, MfB significantly outperforms SfB at bounces, especially in terms of deblurring quality metric PSNR. The performance gap is still significant when evaluating on frames that are adjacent to the bounce but is relatively small when averaged over the whole dataset. This indicates that bounces are significantly more difficult than other parts of the dataset, as shown qualitatively in Fig. 4 and Fig. 3, and our method successfully reconstructs such frames as well. For single-frame approaches, the difficulty comes mainly from the trajectory non-linearity, slight object deformation, and shadows near the bounce point. Motion-from-Blur is robust to these difficulties since the optimization is more constrained from easier frames before and after the bounce, and the trajectory is explicitly modeled with a bounce. On frames that are far from the bounce, the difference in deblurring quality between the single-frame and multi-frame approaches is marginal on the TbD-3D dataset. Note that our model is generic and estimates continuously connected trajectories also if there is no bounce.

Synthetic 3D dataset. We construct a synthetic dataset of fast moving objects with ground-truth 3D models and 3D motions for evaluation. We sample random 3D models from the ShapeNet dataset [1], random linear 3D translations and 3D rotations (for a fair comparison with SfB [31] that reconstructs only linear motions), and random consecutive frames from the VOT [16] tracking dataset as backgrounds. 3D translation is randomly chosen in the interval between 1 to 5 object sizes, and 3D rotation is randomly chosen up to 30° (first subset) or 90° (second subset) during the video duration. Then, we apply the video formation model (6) with $N = 3$ to create two subsets, each consisting of 30 short videos. We report the mesh error e_Θ as the average bidirectional distance between the closest vertices of the ground-truth and the estimated mesh, both placed at the ground-truth and predicted initial 6D pose, and divided by the object size. For evaluating the translation error $e_\mathbf{t}$, we compute the norm of the difference vector between the predicted and ground-truth translation offset $\mathcal{T}(1) - \mathcal{T}(0)$, divided by the object size. Thus, these two scores (e_Θ and $e_\mathbf{t}$) are reported as a fraction of the object size. For evaluating the rotation error $e_\mathbf{r}$, we compute the average angle between the estimated rotation change (rotation between $\mathcal{Q}(1)$ and $\mathcal{Q}(0)$) and the ground-truth one.

We compare to the only other method that can reconstruct a 3D object and its motion from the motion-blurred input (SfB [31]). Our method is applied to all three video frames in each video, whereas SfB is applied to them individually, and the scores are averaged (w.r.t. one video frame). As shown in Table 3, on the synthetic dataset [1] with up to 90° rotation, our method is almost twice as accurate as SfB in terms of 3D translation and 3D rotation estimation. For smaller rotations up to 30°, the difference is smaller but is still significant. This highlights that multi-frame optimization is especially beneficial for complex objects (as from ShapeNet) with non-negligible rotations.

[1] All figures show only real data.

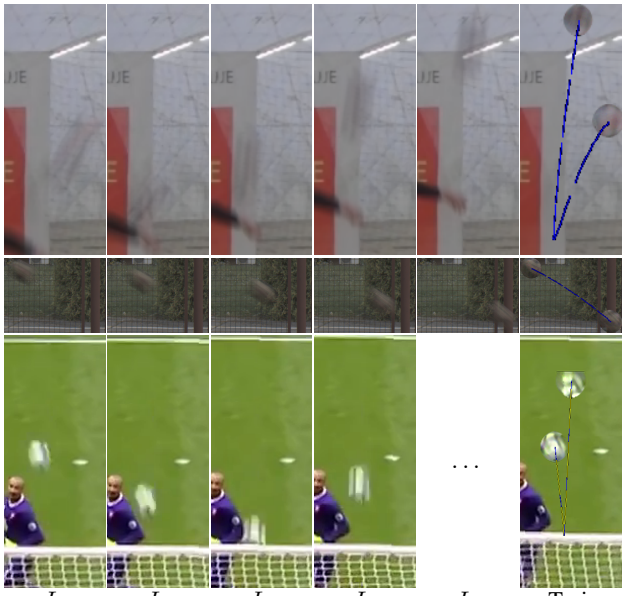

Figure 4. **Reconstructing 2D object trajectories with bounces.** For each video, we reconstruct 3D object and its motion (*blue*: observed trajectory, *yellow*: the exposure gap). We visualize the trajectory for the center of mass of the mesh silhouettes and further render the first and last pose of the object (right-most image). **Top row:** Scene from TbD [14] dataset; **Center row:** TbD-3D [28] scene; **Bottom row:** YouTube scene from Fig. 2.

Applications. MfB can be used for imitating high-speed cameras or multiplying their capabilities by creating temporal super-resolution from motion-blurred videos. MfB can perform 3D reconstruction of blurred objects that are almost unidentifiable by humans, *e.g.* image forensics of surveillance cameras. Applications also include 6D object tracking and reconstruction in sports, *e.g.* football, tennis, basketball.

5. Limitations

Static camera. MfB assumes that the video is captured by a nearly static camera. A moving camera adds even more ambiguity to the observed blur that could stem from both camera and object motion blur. Moreover, motion blur also has to be compensated by the camera motion, and the whole problem would become much more difficult. Since all previous methods for fast moving object deblurring and 3D reconstruction [14, 26, 28, 30, 31] also assume a static camera, tackling this problem remains challenging future work.

Shutter. We assume that the shutter speed is constant. However, some cameras have an adjustable shutter that

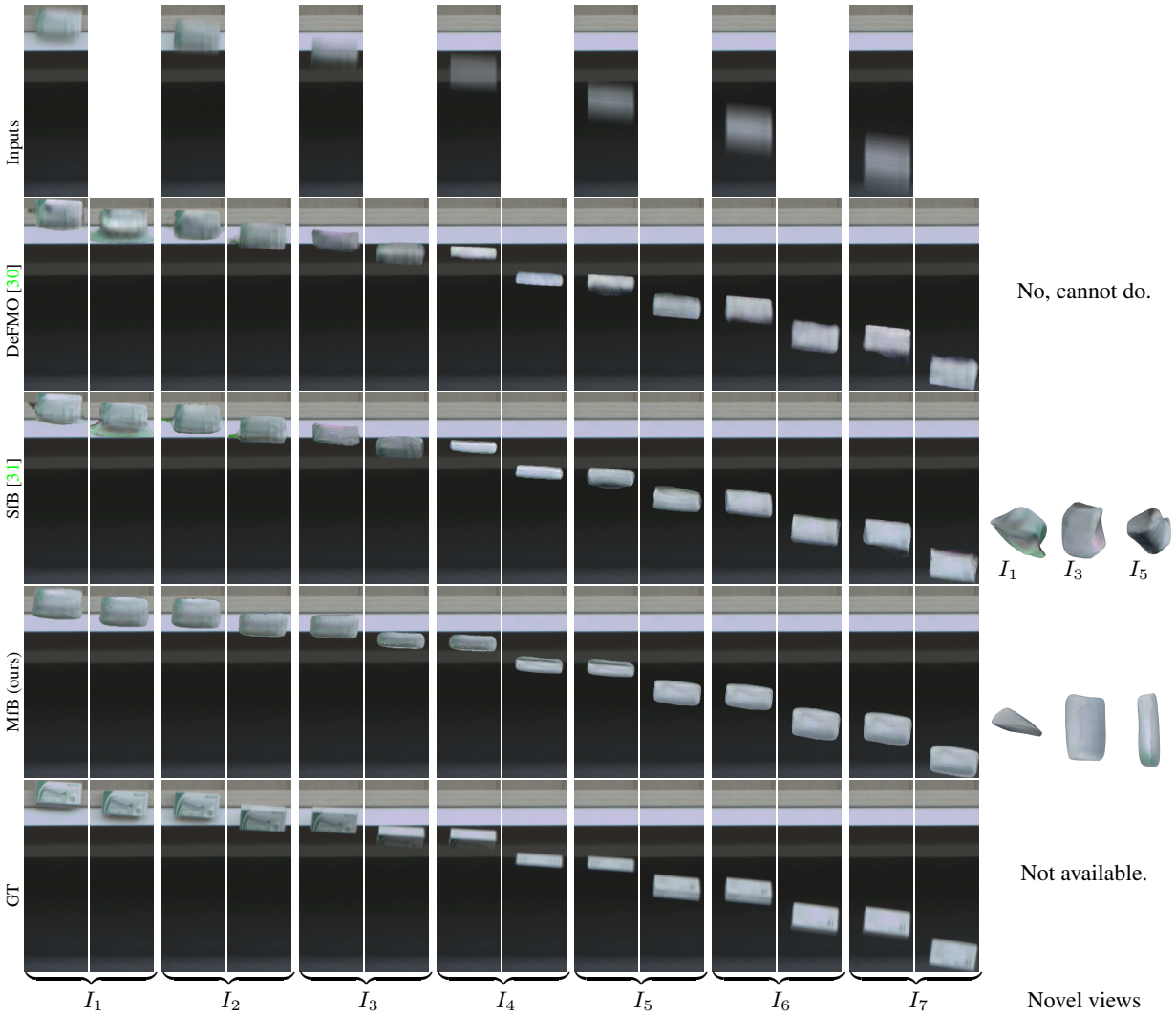

Figure 5. **3D reconstruction and temporal super-resolution of a falling box from Falling Objects dataset [13].** Our method produces more consistent results over $N = 7$ input frames than previous methods and does not suffer from artifacts on frames with shadows (I_1 and I_2). The final 3D reconstruction is also more complete and accurate than the single-frame approach SfB [31] as shown on novel views.

changes the exposure gap based on lighting conditions, *e.g.* less exposure for bright scenes and more exposure for dark scenes. In most cases, this transition is smooth, and our method is robust thanks to the sliding window approach. Modeling a rolling shutter is beyond the scope of this paper.

Texture-less objects. Reconstructing 3D objects that lack noticeable texture is a challenge even for generic 3D reconstruction since no distinctive geometry features are observable, and the correspondences are ambiguous. In this case, detecting any 3D rotation is almost infeasible.

Non-rigid objects. We assume that the object is rigid, *i.e.* its 3D model is constant for the video duration. Such assumption is invalid for deforming objects, which often happens during the bounce. However, since these deformations are often insignificant and only for a very short duration of time, our modeling still handles such cases well.

6. Conclusion

We presented the first method for estimating textured 3D shapes and complex motions of motion-blurred objects in videos. By optimizing over multiple input frames, we correctly recover 3D object shape and motion, its motion direction, and the camera exposure gap. Various experiments have shown that our method produces sharper and more consistent results compared to other methods for fast moving object deblurring. Compared to single-image 3D shape and motion estimation [31], which is a special instance of our approach, we recover more complete shapes and significantly more precise motion estimation.

Acknowledgements. This research was supported by a Google Focused Research Award, Innosuisse grant No. 34475.1 IP-ICT, Research Center for Informatics (project CZ.02.1.01/0.0/0.0/16_019/0000765 funded by OP VVV), and a research grant by FIFA.

References

[1] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 3, 7

[2] Wenzheng Chen, Jun Gao, Huan Ling, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *NeurIPS*, 2019. 3, 4

[3] Zhixiang Chi, Yang Wang, Yuanhao Yu, and Jin Tang. Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9137–9146, June 2021. 2

[4] Tianyu Ding, Luming Liang, Zhihui Zhu, and Ilya Zharkov. Cdfi: Compression-driven network design for frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8001–8011, June 2021. 2

[5] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, July 2017. 3

[6] Shurui Gui, Chaoyue Wang, Qihua Chen, and Dacheng Tao. Featureflow: Robust video interpolation via structure-to-texture generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[7] Muzhi Han, Zeyu Zhang, Ziyuan Jiao, Xu Xie, Yixin Zhu, Song-Chun Zhu, and Hangxin Liu. Reconstructing interactive 3d scenes by panoptic mapping and cad model alignments. In *ICRA*, 2021. 1

[8] Krishna Murthy Jatavallabhula, Edward Smith, Jean-Francois Lafleche, Clement Fuji Tsang, Artem Rozantsev, Wenzheng Chen, Tommy Xiang, Rev Lebaredian, and Sanja Fidler. Kaolin: A pytorch library for accelerating 3d deep learning research. *arXiv:1911.05063*, 2019. 5

[9] Meiguang Jin, Zhe Hu, and Paolo Favaro. Learning to extract flawless slow motion from blurry videos. In *CVPR*, June 2019. 2

[10] Meiguang Jin, Givi Meishvili, and Paolo Favaro. Learning to extract a video sequence from a single motion-blurred image. In *CVPR*, June 2018. 2, 6

[11] Adam Kaufman and Raanan Fattal. Deblurring using analysis-synthesis networks pair. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015. 4

[13] J. Kotera, J. Matas, and F. Šroubek. Restoration of fast moving objects. *IEEE TIP*, 29:8577–8589, 2020. 2, 4, 5, 6, 8

[14] J. Kotera, D. Rozumnyi, F. Šroubek, and J. Matas. Intra-frame object tracking by deblatting. In *ICCVW*, Oct 2019. 2, 5, 6, 7

[15] J. Kotera and F. Šroubek. Motion estimation and deblurring of fast moving objects. In *ICIP*, pages 2860–2864, Oct 2018. 2

[16] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomas Vojir, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE TPAMI*, 38(11):2137–2155, Nov 2016. 7

[17] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[18] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *ICCV*, Oct 2019. 2, 6

[19] Dongxu Li, Chenchen Xu, Kaihao Zhang, Xin Yu, Yiran Zhong, Wenqi Ren, Hanna Suominen, and Hongdong Li. Arvo: Learning all-range volumetric correspondence for video deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7721–7731, June 2021. 2

[20] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 3

[21] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *CVPR*, 2020. 2

[22] Jinshan Pan, Haoran Bai, and Jinhui Tang. Cascaded deep video deblurring using temporal sharpness prior. In *CVPR*, June 2020. 2

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[24] Stephan R. Richter and Stefan Roth. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *CVPR*, June 2018. 3

[25] D. Rozumnyi, J. Kotera, F. Šroubek, and J. Matas. Non-causal tracking by deblatting. In Gernot A. Fink, Simone Frintrop, and Xiaoyi Jiang, editors, *GCPR*, pages 122–135, Cham, 2019. Springer International Publishing. 2

[26] D. Rozumnyi, J. Kotera, F. Šroubek, and J. Matas. Tracking by deblatting. *IJCV*, 129(9):2583–2604, 2021. 2, 4, 7

[27] D. Rozumnyi, J. Kotera, F. Šroubek, L. Novotný, and J. Matas. The world of fast moving objects. In *CVPR*, pages 4838–4846, July 2017. 2

[28] D. Rozumnyi, J. Kotera, F. Šroubek, and J. Matas. Sub-frame appearance and 6d pose estimation of fast moving objects. In *CVPR*, pages 6777–6785, 2020. 2, 5, 6, 7

[29] Denys Rozumnyi, Jiří Matas, Filip Šroubek, Marc Pollefeys, and Martin R. Oswald. Fmodetect: Robust detection of fast moving objects. In *ICCV*, pages 3541–3549, October 2021. 2

[30] Denys Rozumnyi, Martin R. Oswald, Vittorio Ferrari, Jiri Matas, and Marc Pollefeys. Defmo: Deblurring and shape recovery of fast moving objects. In *CVPR*, Nashville, Tennessee, USA, Jun 2021. 2, 4, 5, 6, 7, 8

[31] Denys Rozumnyi, Martin R. Oswald, Vittorio Ferrari, and Marc Pollefeys. Shape from blur: Recovering textured 3d shape and motion of fast moving objects. In *NeurIPS*, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[32] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 3

[33] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 1, 3

[34] Wang Shen, Wenbo Bao, Guangtao Zhai, Li Chen, Xiongkuo Min, and Zhiyong Gao. Blurry video frame interpolation. In *CVPR*, June 2020. 2

[35] Li Siyao, Shiyu Zhao, Weijiang Yu, Wenxiu Sun, Dimitris Metaxas, Chen Change Loy, and Ziwei Liu. Deep animation video interpolation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6587–6595, June 2021. 2

[36] Władysław Skarbek. Shape from motion revisited. In Dominik Ślezak, Gerald Schaefer, Son T. Vuong, and Yoo-Sung Kim, editors, *Active Media Technology*, pages 383–394, Cham, 2014. Springer International Publishing. 1, 3

[37] Jan Stuhmer, Sebastian Nowozin, Andrew Fitzgibbon, Richard Szeliski, Travis Perry, Sunil Acharya, Daniel Cremers, and Jamie Shotton. Model-based tracking at 300hz using raw time-of-flight observations. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2

[38] Maitreya Suin, Kuldeep Purohit, and A. N. Rajagopalan. Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[39] Maxim Tatarchenko*, Stephan R. Richter*, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *CVPR*, 2019. 1, 3

[40] F. Šroubek and J. Kotera. Motion blur prior. In *ICIP*, pages 928–932, 2020. 2

[41] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, 2018. 3, 4

[42] Yi Xu, Yuzhang Wu, and Hui Zhou. Multi-scale voxel hashing and efficient 3d representation for mobile augmented reality. In *CVPRW*, June 2018. 1

[43] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 1, 3

[44] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *CVPR*, June 2020. 2

[45] Aleš Zita and Filip Šroubek. Tracking fast moving objects by segmentation network. In *ICPR*, pages 10312–10319, 2021. 2