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

Denys Rozumnyi[1,4] Martin R. Oswald[3,1] Vittorio Ferrari[2] Marc Pollefeys[1]
[1]Department of Computer Science, ETH Zurich [2]Google Research Zurich
[3]University of Amsterdam [4]Czech Technical University in Prague

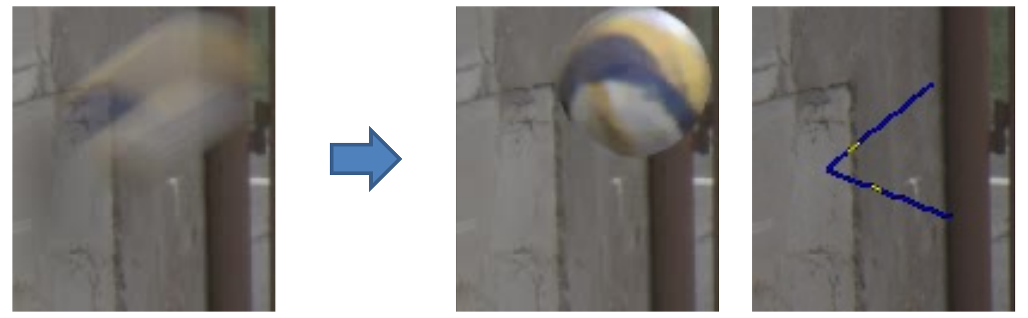
CVPR JUNE 19-24 2022 NEW ORLEANS • LOUISIANA

Introduction

We jointly estimate the 3D motion, 3D shape, and texture of motion-blurred objects by optimizing over multiple motion-blurred frames.

Contributions:

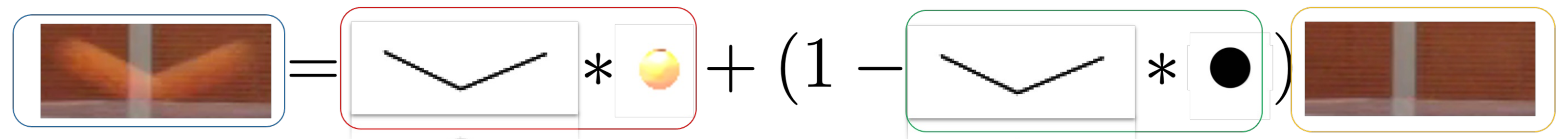
- First method to optimize over a video instead of a single frame [1-5].
- Better motion modeling, e.g., with bounces and acceleration.
- Explicit exposure gap modeling and its automatic estimation.



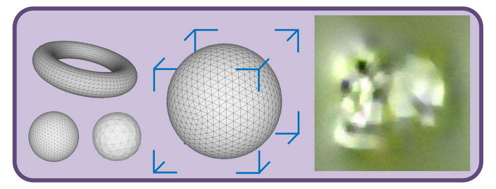
Classical motion blur problem formulation for a single frame:

Blurred object Foreground Visibility map Background

$$I = H * F + (1 - H * M)\, B$$

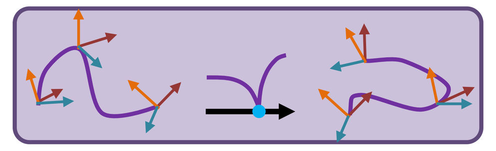


Modeling



Mesh Θ: prototype index, vertex offsets, texture map.

Fixed mesh parameters: faces, initial vertex positions, texture mapping.



Motion Ω: continuous 3D translation and 3D rotation.

They are modelled by piece-wise polynomials to allow for bounces (abrupt motion changes), acceleration, and other forces.

Exposure gap ϵ: real-valued parameter that denotes fraction of time when camera shutter is closed.

We estimate it automatically as part of the proposed optimization.

Video formation with motion-blurred objects

Many frames Exposure gap Motion function More complex motion

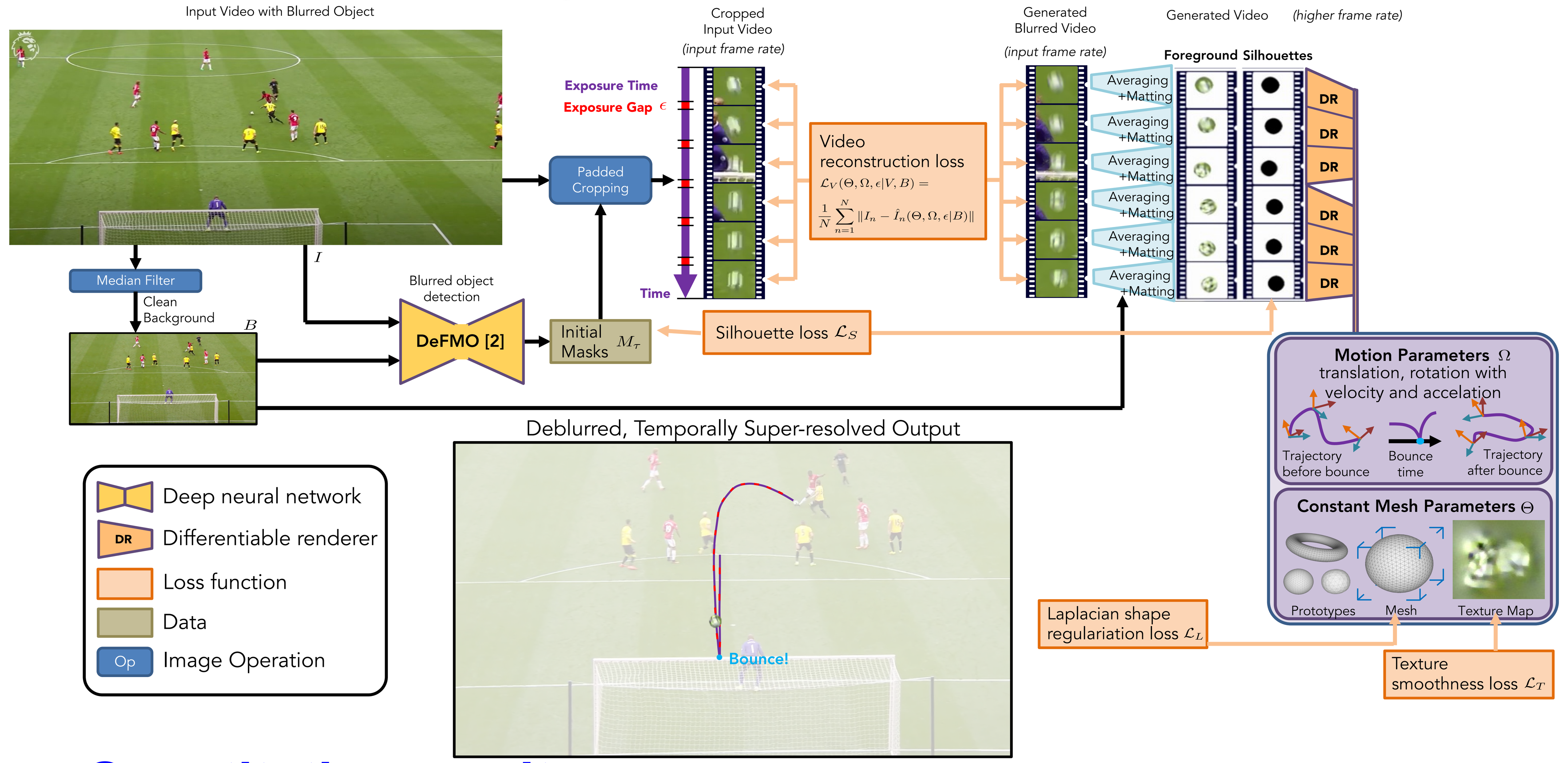
$$\hat{I}_n(\cdot) = \int_{\frac{n-1}{N}}^{\frac{n-\epsilon}{N}} \mathcal{R}_F\Big(\mathcal{M}\big(\Theta, \mathcal{Q}(\tau), \mathcal{T}(\tau)\big)\Big)\mathrm{d}\tau + \Bigg(1 - \int_{\frac{n-1}{N}}^{\frac{n-\epsilon}{N}} \mathcal{R}_S\Big(\mathcal{M}\big(\Theta, \mathcal{Q}(\tau), \mathcal{T}(\tau)\big)\Big)\mathrm{d}\tau\Bigg)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

Motion blur (averaging) Differentiable rendering Shared parameters across frames

- If we knew values of all parameters, we could render the input video!
- Strict generalization of [1], who did it for a single frame and linear motion.

Model fitting

- All parameters are optimized to re-render the input video as closely as possible by minimizing the pixel-wise reprojection error using differentiable rendering [6].



Quantitative results

- New state-of-the-art on fast moving object deblurring benchmark [2].
- Compared to a single-frame approach [1]:

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

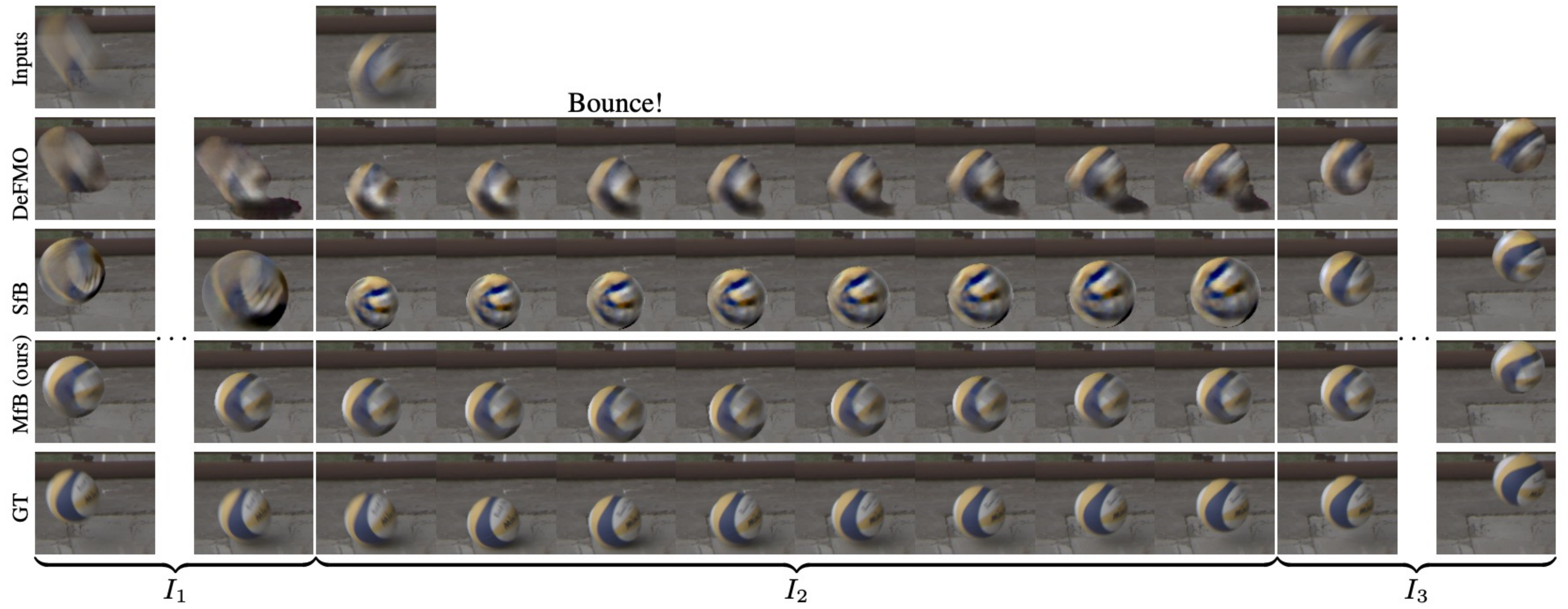
Table 1: deblurring quality at bounces.

		$e_{\mathbf{t}}\downarrow$	$e_{\mathbf{r}}\downarrow$	$e_{\Theta}\downarrow$
$< 90^\circ$	SfB [1]	37.8 %	10.9°	3.0 %
	MfB (ours)	**20.0 %**	**6.4°**	**2.7 %**
$< 30^\circ$	SfB [1]	12.8 %	4.8°	2.3 %
	MfB (ours)	**8.8 %**	**3.7°**	**2.2 %**

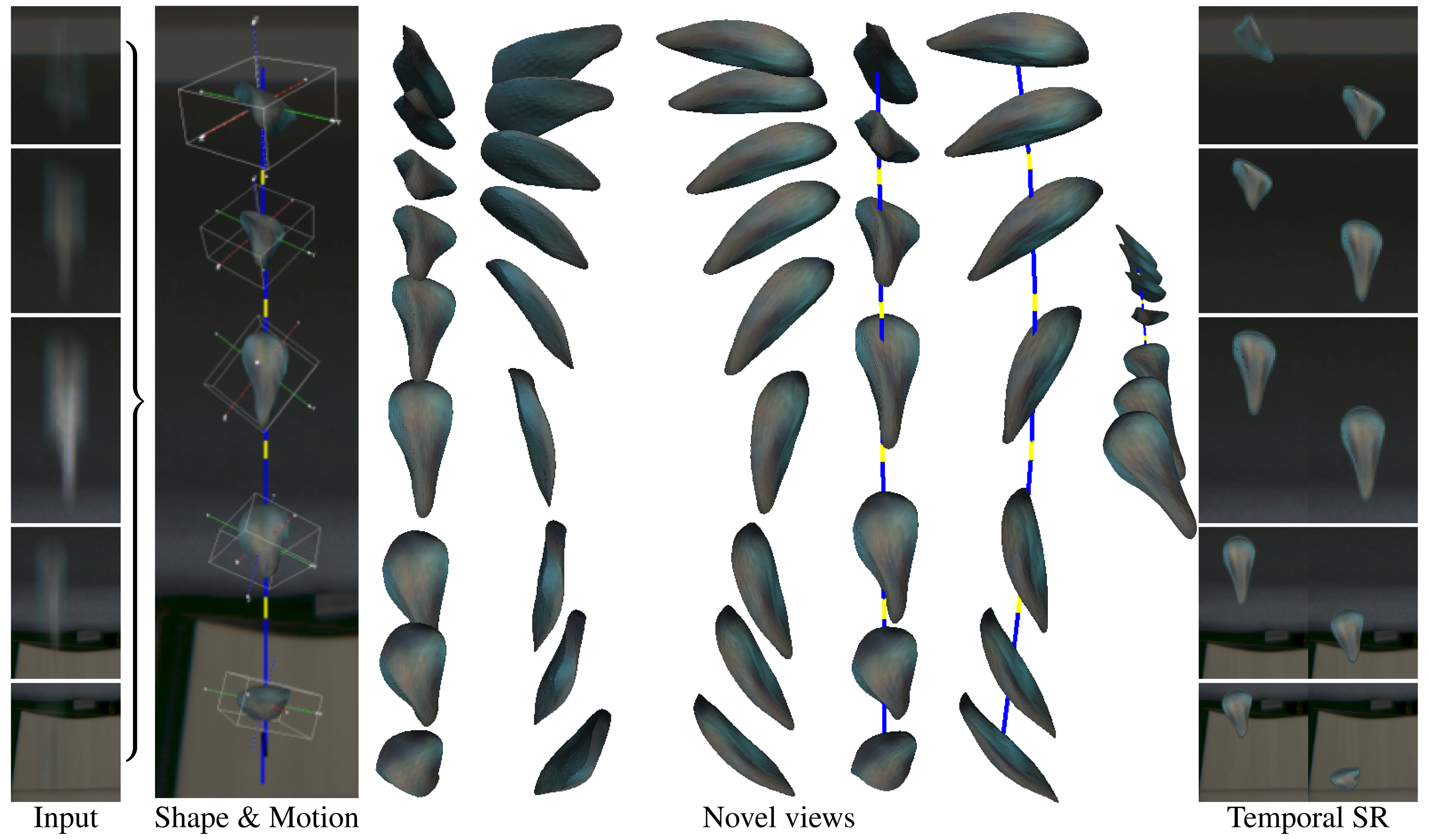
Table 2: evaluating 3D translation, 3D rotation, and 3D shape on a synthetic dataset.

Qualitative results

- Significantly better performance at bounces:



- More complete 3D reconstruction, especially on back sides.
- Bounce detection and sharper deblurring.



Implementation and more results on GitHub:

References

[1] D. Rozumnyi et al.: Shape from Blur @ NeurIPS 2021
[2] D. Rozumnyi et al.: DeFMO @ CVPR 2021
[3] D. Rozumnyi et al.: FMODetect @ ICCV 2021
[4] D. Rozumnyi et al.: TbD-3D @ CVPR 2020
[5] J. Kotera et al.: Restoration of Fast Moving Objects @ TIP 2020
[6] W. Chen et al.: DIBR @ NeurIPS 2019