[Supplementary material]: Joint 3D Shape and Motion Estimation from Rolling Shutter Light-Field Images

Hermes McGriff [1,3] Renato Martins[1,2] Nicolas Andreff [3] Cédric Demonceaux[1,2]
[1]Université de Bourgogne, CNRS UMR 6303 ICB [2]Université de Lorraine, CNRS, Inria, LORIA
[3]Université de Franche-Comté, CNRS UMR 6174 FEMTO-ST
{hermes.mc-griff,renato.martins,cedric.demonceaux}@u-bourgogne.fr, nicolas.andreff@univ-fcomte.fr

In this supplementary material to our paper, we provide additional visualizations of the sequences from our dataset with rolling shutter light-field images, as well as more details about the projection model presented in the paper.

1. Rolling Shutter Light-Field Dataset

Visualizations of some sequences scenes from our dataset (discussed in Section 4) are shown in Fig. 1. For each scene, we provide eleven velocity scenarios from which every middle-exposition time position is similar, *i.e*. the center line of pixel is the same for every image of the same scene. All the scenes share the same eleven velocity profiles, but still possess different deformations due to the RS effect since the center of rotation of the camera is always different. Please notice the far right column in Fig. 1 where the same movement creates some "squishing" in some sequences (*e.g*. second and fourth rows) and some "stretching" in others (*e.g*. third and fifth rows). The velocities for each scenarios are presented in Tab. 1.

Scenario number	1	2	3	4	5
Rotations (euler angles)	[0,0,$\pi/12$]	[0,0,0]	[-$\pi/18$, 0, 0]	[$\pi/18$, $\pi/18$, 0]	[0,0,$\pi/12$]
Translations	[0,0,0]	[0,-0.2,0]	[0,-0.05,0.05]	[0,0,0.2]	[0,-0.2,0]
Scenario number	6	7	8	9	10
Rotations (euler angles)	[0,0,$\pi/3$]	[0,0,0]	[-$\pi/3$, 0, 0]	[$2\pi/9$, 0, 0]	[0,0,$\pi/2$]
Translations	[0,0,0]	[0,-0.8,0]	[0,0.4,0.2]	[0.4,-1.6,-0.8]	[0,-0.8,0]

Table 1. Different velocity scenarios, given in radians per frames and meters per frames. The scenarios from 1 to 5 are the *slow* scenarios and the ones from 6 to 10 are the *fast* scenarios.

All sequences were created using Blender and the render engine Cycles. The cameras have a $50mm$ focal length, and they are placed in a plane normal to the z axis organized in a 9×9 grid with $6mm$ between each view point. They are all oriented in a way that they have their optical axes passing through the point $[0, 0, 7m]$. We also provide $1024x1024$ depth maps generated from the camera placed at the same position that of the center view, but with a $25mm$ focal length. Since the scene is moving with respect to the camera, it is necessary to have a wider field of view as more of the scene can be seen in the RS scenarios. The depth maps are normalized for a range from $0m$ to $7m$. Some vizualisations of the depth maps are shown in the second column of Fig. 1. The detailed results mentioned in the main paper for all sequences and scenes are presented Tab. 2.

2. Projection Model Formulation

We provide additional details of the construction of the formulation for the RSLF projection model presented in Section 3, until Eq. (5), of the main paper. In order to describe the projection of a point in the world coordinate frame, we first apply a thin lens projection through the main lens and then a pinhole projection through every micro-lens of the micro-lens array (MLA) independently (as shown in Fig. 2). Given a point in the world homogeneous coordinate frame ${}^{w}\tilde{\mathbf{p}} =$

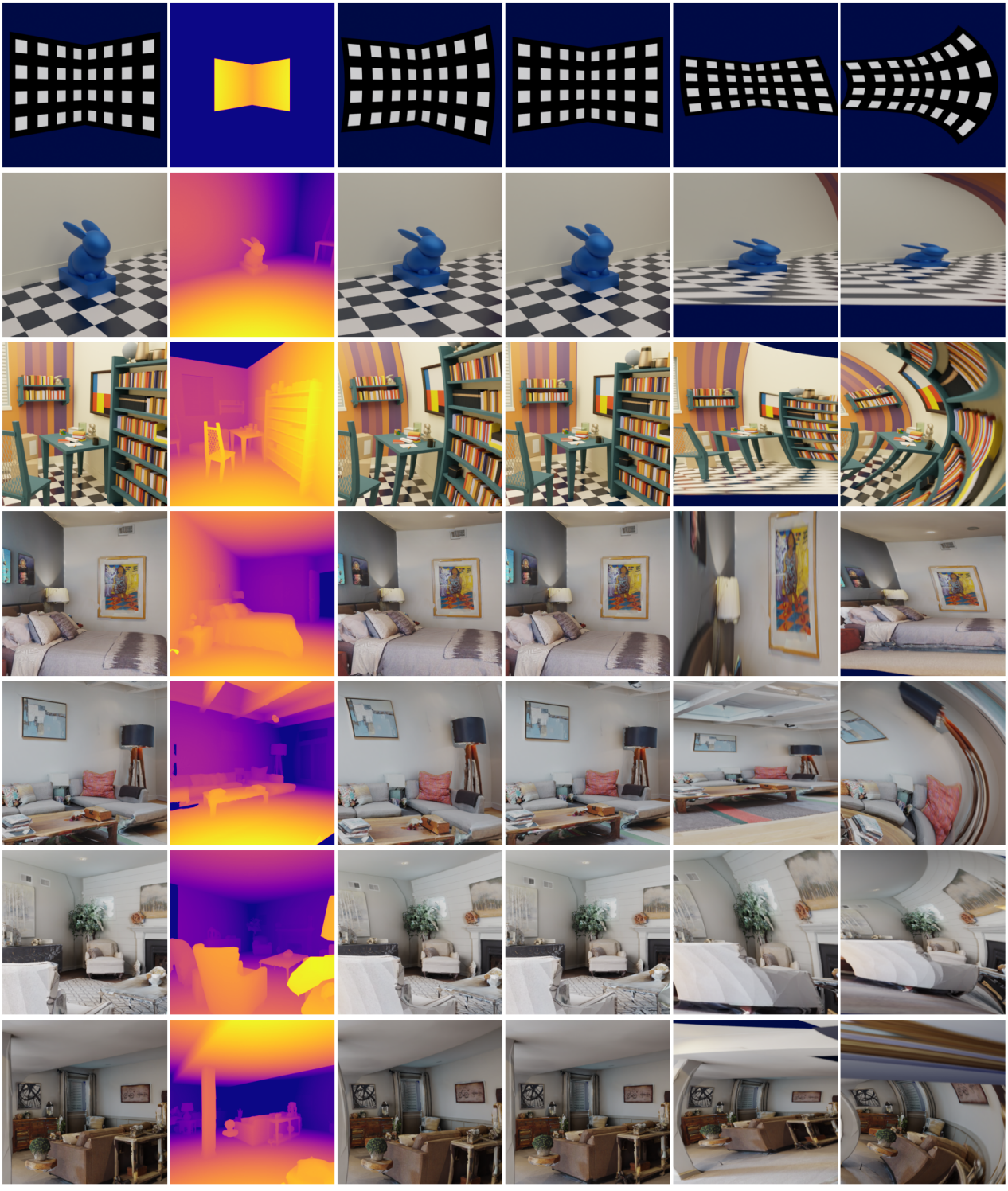

Figure 1. This figure is already presented in the main paper and is duplicated here with bigger dimensions - Visualizations of the center views from each of the seven different scenes. The four bottom rows are generated from samples of the Habitat-Matterport benchmark [3]. The first column shows the GS scenario; the second column shows the associated depth maps, and the subsequent columns are from different motion scenarios (numbers 1, 2, 9 and 10 in Tab. 1).

	abs rel ↓											$\delta < 1.25$ ↑										
rabbit	0	1	2	3	4	5	6	7	8	9	10	0	1	2	3	4	5	6	7	8	9	10
Jeon-CVPR [2]	0.06	0.08	0.07	0.07	0.07	0.1	0.19	0.12	0.13	0.34	0.39	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.82	**1.0**	0.91	0.59	0.35
OACC-Net [4]	0.4	0.48	0.5	0.44	0.38	0.49	0.47	0.48	0.44	0.5	0.5	0.26	0.08	0.06	0.14	0.29	0.09	0.1	0.1	0.1	0.13	0.1
Ours	**0.03**	**0.03**	**0.03**	**0.02**	**0.03**	**0.03**	**0.03**	**0.03**	**0.02**	**0.03**	**0.03**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
table	0	1	2	3	4	5	6	7	8	9	10	0	1	2	3	4	5	6	7	8	9	10
Jeon-CVPR [2]	0.03	0.03	0.04	0.03	0.05	0.03	0.05	0.09	0.05	0.17	0.07	**1.0**	**1.0**	0.99	**1.0**	**1.0**	1.0	0.97	0.96	0.99	0.76	0.94
OACC-Net [4]	0.17	0.21	0.2	0.19	0.19	0.2	0.19	0.24	0.15	0.25	0.2	0.69	0.6	0.64	0.64	0.63	0.59	0.67	0.55	0.79	0.5	0.65
Ours	**0.02**	**0.02**	**0.02**	**0.03**	**0.02**	**0.03**	**0.04**	**0.02**	**0.04**	**0.03**	**0.04**	0.995	0.995	**0.99**	**1.0**	0.99	**1.0**	**1.0**	**0.99**	**0.99**	**0.98**	**1.0**
bedroom	0	1	2	3	4	5	6	7	8	9	10	0	1	2	3	4	5	6	7	8	9	10
Jeon-CVPR [2]	**0.02**	0.03	**0.02**	0.04	0.06	0.03	0.07	**0.02**	0.11	**0.03**	0.07	**1.0**	1.0	**1.0**	**1.0**	0.97	1.0	0.99	**1.0**	0.89	**1.0**	0.94
OACC-Net [4]	0.03	0.05	0.03	0.06	0.1	0.05	0.13	0.03	0.13	0.05	0.13	1.0	0.98	1.0	0.99	0.93	0.98	0.8	1.0	0.77	**1.0**	0.79
Ours	0.03	**0.03**	0.03	**0.03**	**0.03**	**0.03**	**0.04**	0.03	**0.04**	0.04	**0.05**	**1.0**	**1.0**	**1.0**	0.999	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**0.99**
couch	0	1	2	3	4	5	6	7	8	9	10	0	1	2	3	4	5	6	7	8	9	10
Jeon-CVPR [2]	**0.02**	**0.03**	**0.03**	**0.03**	**0.03**	**0.02**	**0.06**	0.05	**0.06**	0.05	**0.06**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**0.98**	**1.0**
OACC-Net [4]	0.04	0.06	0.06	0.05	0.06	0.05	0.07	0.09	0.06	0.12	0.06	1.0	0.99	0.99	1.0	0.98	0.99	0.99	0.92	**1.0**	0.84	0.98
Ours	0.03	0.04	0.03	0.03	0.03	0.04	0.06	**0.03**	0.11	**0.04**	0.06	**1.0**	0.999	**1.0**	0.999	0.989	0.999	0.989	0.992	0.964	0.958	0.971
fireplace	0	1	2	3	4	5	6	7	8	9	10	0	1	2	3	4	5	6	7	8	9	10
Jeon-CVPR [2]	**0.05**	**0.08**	**0.08**	**0.09**	**0.1**	0.09	0.09	0.1	0.32	0.09	0.14	**0.96**	**0.9**	**0.87**	0.9	**0.86**	**0.84**	**0.85**	**0.89**	0.57	0.86	0.71
OACC-Net [4]	0.08	0.13	0.14	0.11	0.13	0.15	0.14	0.16	0.35	0.14	0.15	0.95	0.86	0.81	**0.9**	0.84	**0.84**	0.85	0.86	**0.58**	0.84	**0.8**
Ours	0.12	0.1	0.12	0.13	0.1	**0.09**	**0.09**	**0.09**	**0.32**	**0.07**	**0.13**	0.741	0.778	0.751	0.716	0.8	0.837	0.828	0.875	0.558	**0.92**	0.753
living$_r oom$	0	1	2	3	4	5	6	7	8	9	10	0	1	2	3	4	5	6	7	8	9	10
Jeon-CVPR [2]	0.04	0.04	0.05	0.04	**0.05**	**0.05**	**0.06**	0.07	0.07	0.1	0.1	0.99	1.0	**0.99**	0.99	**0.96**	0.99	**0.99**	0.95	0.99	**0.92**	0.94
OACC-Net [4]	**0.04**	**0.04**	0.05	0.04	0.08	0.05	0.07	0.08	**0.06**	0.12	0.09	**1.0**	**1.0**	0.99	**0.99**	0.96	**0.99**	0.99	**0.95**	**1.0**	0.87	**0.96**
Ours	0.05	0.05	**0.04**	**0.04**	0.07	0.05	0.08	**0.06**	0.12	**0.08**	**0.07**	0.974	0.98	0.983	0.974	0.93	0.975	0.92	0.946	0.928	0.915	0.878

Table 2. Detailed reconstruction error metrics for all the scenes considering the eleven different motion scenarios (from 0 to 10) of the dataset. The upward arrow means that a higher score is better.

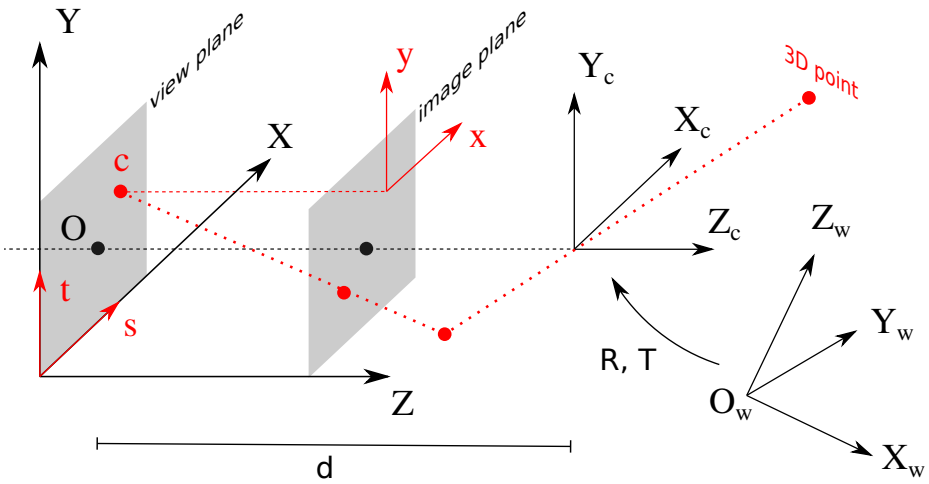

Figure 2. This figure is already presented in the main paper and is duplicated here for the convenience of the reader - The adopted LF coordinate frames: The 3D point is projected in a 3D virtual scene by thin lens projection, then on the 2D image plane by pinhole projection which coordinate frame depends on the considered viewpoint.

$(x_w, y_w, z_w, 1)^\top$, its coordinates in the camera coordinate frame will be given by the matrix ${}^c\mathbf{M}_w{}^w$ as

$$ {}^c\widetilde{\mathbf{p}} = {}^c\mathbf{M}_w{}^w\widetilde{\mathbf{p}}, \tag{1} $$

where $\widetilde{\mathbf{p}}$ represents the homogeneous coordinates of point $\mathbf{p}$ in the left-upperscripted reference frame, while ${}^c\mathbf{M}_w$ is the homogeneous matrix displacing the camera frame onto the world frame:

$$ {}^c\mathbf{M}_w = \begin{bmatrix} {}^c\mathbf{R}_w & {}^c\mathbf{T}_w \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{2} $$

We will then project it through the main lens as ${}^v\widetilde{\mathbf{p}} = (x_v, y_v, z_v, 1)^\top$, using the projection matrix for a thin lens projection

as

$$\lambda_c {}^v\widetilde{\mathbf{p}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{F} & 1 \end{bmatrix}}_{=:\mathbf{K}_c} {}^c\widetilde{\mathbf{p}}, \tag{3}$$

with F the focal distance of the main lens and λ_c a scaling factor. To express this point $\widetilde{\mathbf{p}} = (x, y, z, 1)^\top$ in the sensor coordinate frame, we use the homogeneous matrix $\mathbf{D}$ linked to the geometry of the sensor as

$$\widetilde{\mathbf{p}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & O_x \\ 0 & 1 & 0 & O_y \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{=:\mathbf{D}} {}^v\widetilde{\mathbf{p}}, \tag{4}$$

with $\mathbf{O} = (O_x, O_y, 0)^\top$ the intersection of the optical axis and the view plane and d the distance between the optical center of the main lens and the view plane.
Given a point $\mathbf{c} = (s, t, 0)^\top$ from the view plane, *i.e.* a projection center, the pinhole projection to an image point $\widetilde{\mathbf{m}}^{s,t} = (x^{s,t}, y^{s,t}, 1)^\top$ is given by the matrix

$$\lambda_s^{s,t} \widetilde{\mathbf{m}}^{s,t} = \underbrace{\begin{bmatrix} f & 0 & 0 & -fs \\ 0 & f & 0 & -ft \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{=:\mathbf{K}_s^{s,t}} \widetilde{\mathbf{p}}, \tag{5}$$

with f the focal distance of the micro-lenses, *i.e.* the distance between the view plane and the image plane and λ_s a scaling factor. This is a classical pinhole projection that take into account the position of the micro-lens $\mathbf{c}$. From these equations we can find the LF point (x, y, s, t) for any 3D point in the world, in the case of a LF camera modeled with the GS hypothesis. In order to add the RS effect in our model, and thus the movement of the camera in our equations, we adopt a similar RS formalism from Ait-Aider *et al.* [1]. They considered that the camera moves by the same little uniform movement between any two lines of pixels and define camera pose in function of the pixel line. We also make the hypothesis that the acquisition time inside a micro-image is instantaneous (*i.e.* micro-images are considered GS). Assuming a uniform movement $[\delta\mathbf{R}^{\delta t} \mid \delta\mathbf{T}^{\delta t}]$ between any two lines of micro-images, we can express the camera pose in function of the viewpoint line and rewrite Eq. (1) as

$${}^c\widetilde{\mathbf{p}} = \begin{bmatrix} \delta\mathbf{R}^t {}^c\mathbf{R}_w & {}^c\mathbf{T}_w + \delta\mathbf{T}^t \\ \mathbf{0}^\top & 1 \end{bmatrix} {}^w\widetilde{\mathbf{p}}, \tag{6}$$

with

$$\delta\mathbf{R}^t = \mathbf{a}\mathbf{a}^\top(1 - \cos(\Omega\tau t)) + \boldsymbol{I}\cos(\Omega\tau t) + [\mathbf{a}]_\wedge \sin(\Omega\tau t), \text{ and } \delta\mathbf{T}^t = \mathbf{v}\tau t, \tag{7}$$

with $\mathbf{a}$ (axis of rotation) Ω (angular velocity) and $\mathbf{v}$ (linear velocity) describes the uniform movement of the camera coordinate frame with respect to the world coordinate frame. τ is the time between the acquisition of two lines of point of view and t is the line coordinate of the point of view $\mathbf{c} = (s, t, 0)^\top$ from Eq. (5). The complete RSLF projection of the 3D point ${}^w\widetilde{\mathbf{p}}_i$ to a image point $\mathbf{m}_i^{s,t}$, given a center of projection $\mathbf{c} = (s, t, 0)^\top$, is then

$$\lambda\mathbf{m}_i^{s,t} = \mathbf{K}_s^{s,t}\mathbf{D}\mathbf{K}_c[\delta\mathbf{R}^t {}^c\mathbf{R}_w \mid {}^c\mathbf{T}_w + \delta\mathbf{T}^t]^w\widetilde{\mathbf{p}}_i. \tag{8}$$

That can be simplified as

$$\lambda\mathbf{m}_i^{s,t} = \mathbf{K}^{s,t}[\delta\mathbf{R}^t {}^c\mathbf{R}_w \mid {}^c\mathbf{T}_w + \delta\mathbf{T}^t]^w\widetilde{\mathbf{p}}_i, \tag{9}$$

with

$$\mathbf{K}^{s,t} = \begin{bmatrix} f & 0 & -\frac{f}{F}(O_x - s) & f(O_x - s) \\ 0 & f & -\frac{f}{F}(O_y - t) & f(O_y - t) \\ 0 & 0 & 1 - \frac{d}{F} & d \end{bmatrix}, \tag{10}$$

which is presented in Eq. (5) of the main paper. As discussed in the main paper, this projection model combines at the same time the property of a light-field sensor and a rolling shutter sensor. Specifically, the model can be extended to a GS light-field camera, when the temporal delay between two consecutive lines is zero, $\tau = 0$, and thus the position of the sensor with

respect to the scene will be identical for any t. In fact, Eq. (6) will be simplified into Eq. (1), that is the global shutter case of the light-field projection described earlier. For similar reasons, the model will act like a global shutter light-field camera when the camera has no velocity with respect to the object. The RSLF projection model generalizes to a conventional camera projection in the case where the MLA is composed of a unique lens. Indeed if we set $s = 0$ and $t = 0$, Eq. (10) becomes

$$\mathbf{K}^{0,0} = \begin{bmatrix} f & 0 & -\frac{f}{F}O_x & fO_x \\ 0 & f & -\frac{f}{F}O_y & fO_y \\ 0 & 0 & 1 - \frac{d}{F} & d \end{bmatrix} \tag{11}$$

which correspond to a pinhole camera with projection matrix $\mathbf{K}'$ at position $\mathbf{D}'$, with

$$\mathbf{K}' = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{D}' = \begin{bmatrix} 1 & 0 & 0 & O_x - c_x\frac{d}{F} \\ 0 & 1 & 0 & O_y - c_y\frac{d}{F} \\ 0 & 0 & \frac{fO_x}{Fc_x} & d \end{bmatrix}, \tag{12}$$

with $c_x = (\frac{d}{F} - 1)\frac{f}{F}O_x$ and $c_y = (\frac{d}{F} - 1)\frac{f}{F}O_y$. Since we consider that the micro-images are locally global shutter, this pinhole model is global shutter.

Implementation details. The non-linar bundle adjustment discussed in Section 3.1 of the main paper was implemented using PyTorch with the Adam optimizer, with learning rate 0.01 for 5000 iterations to ensure convergence. For the regularization of our optimization method discussed in the end of Section 3 of the main paper, we use a new coordinate frame in order to provide a center of rotation to be optimized and a normalization for the point cloud. We define the new points ${}^n\mathbf{p}_i$ as:

$${}^n\mathbf{p}_i = \frac{\mathbf{p}_i - \mathbf{g}}{\lambda_n}, \tag{13}$$

with $\mathbf{g}$ the center of rotation and λ_n the normalization factor. $\mathbf{g}$ is initialized as the mean position of the initial points $\mathbf{p}_i$ and λ_n is defined before the optimization and is calculated as

$$\lambda_n = \max(\mathbf{p}_i - \mathbf{g}). \tag{14}$$

References

[1] Omar Ait-Aider, Nicolas Andreff, Jean Marc Lavest, and Philippe Martinet. Simultaneous object pose and velocity computation using a single view from a rolling shutter camera. In *Eur. Conf. Comput. Vis.*, 2006. 4

[2] Hae-Gon Jeon, Jaesik Park, Gyeongmin Choe, Jinsun Park, Yunsu Bok, Yu-Wing Tai, and In So Kweon. Accurate depth map estimation from a lenslet light field camera. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015. 3

[3] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied ai. In *Adv. Neural Inform. Process. Syst.*, 2021. 2

[4] Yingqian Wang, Longguang Wang, Zhengyu Liang, Jungang Yang, Wei An, and Yulan Guo. Occlusion-aware cost constructor for light field depth estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 3