Leveraging blur information in plenoptic cameras:

Application to metric depth estimation

Mathieu Labussière[1] Céline Teulière[1] Frédéric Bernardin[2] Omar Ait-Aider[1]

Journée inter-GdR ISIS et Robotique - Capteurs visuels émergents pour la robotique

10 Nov. 2022 - Paris

[1]Université Clermont Auvergne, Clermont Auvergne INP, CNRS, Institut Pascal, F-63000 Clermont-Ferrand, France
[2]Cerema, Équipe-projet STI, 10 rue Bernard Palissy, F-63017 Clermont-Ferrand, France

Contact: mathieu.labu@gmail.com, firstname.name@{uca,cerema}.fr

Goal

Investigate the use of a non-conventional imaging system, the plenoptic camera, for computer vision in robotics applications

Figure 1: `Raytrix R12` plenoptic camera

Table of contents

Background

Background

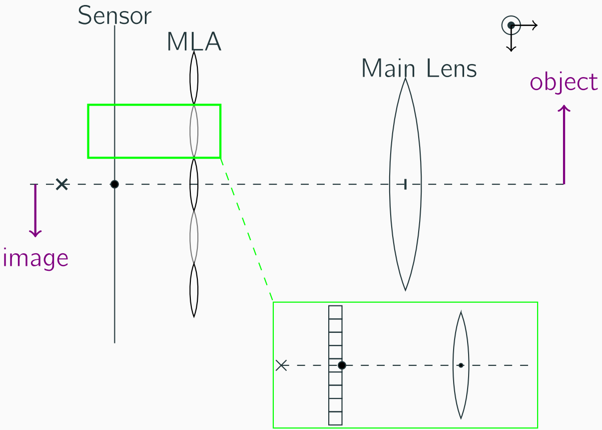
Camera model

(Multi-focus) plenoptic camera

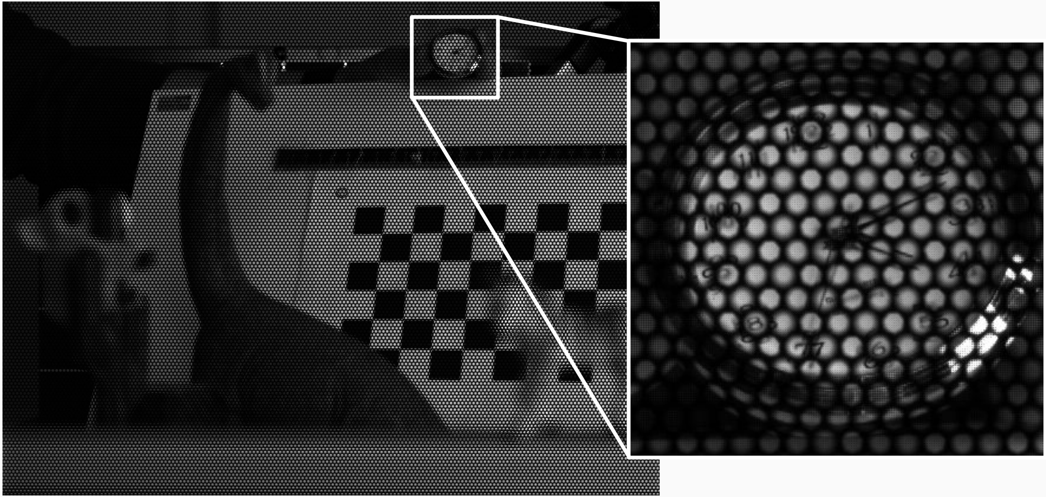
Plenoptic cameras based on a micro-lenses array (MLA) placed between the main lens and the photo-sensible sensor

---

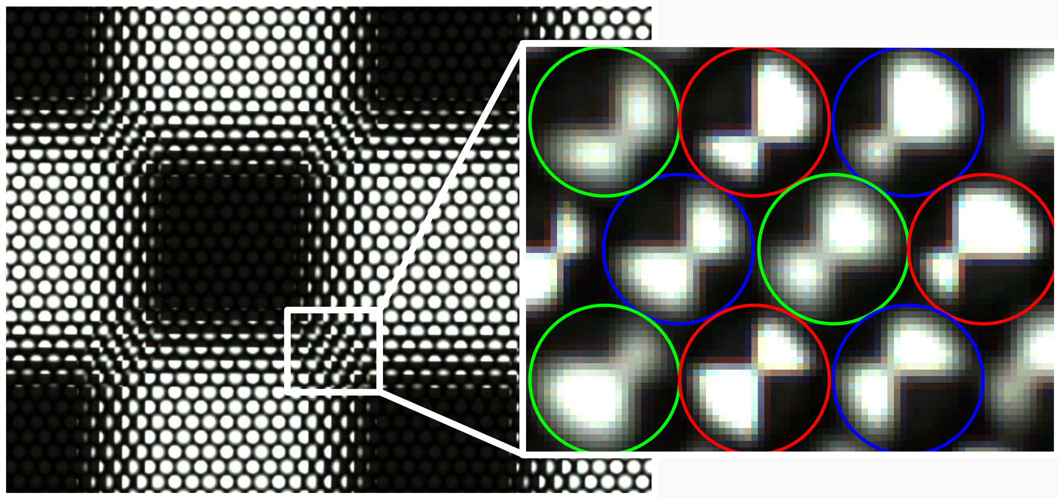
The MLA has different lens types with different focal lengths



(Multi-focus) plenoptic camera



(Multi-focus) plenoptic camera

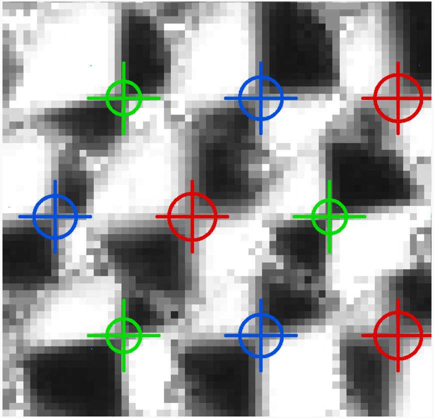


Blur Aware Plenoptic (BAP) Feature

Blurred image of a point = blur circle

Our blur aware plenoptic (BAP) feature [1] is characterized by its center (u, v) and its radius ρ:

$$\boldsymbol{p} = (u, v, \rho)\,. \qquad (1)$$



[1] Labussière et al., "Blur Aware Calibration of Multi-Focus Plenoptic Camera" (*CVPR, 2020*)

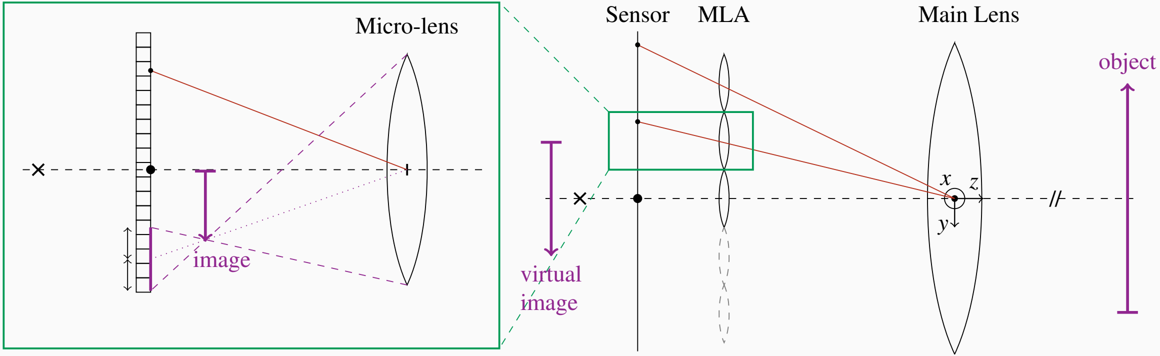
(Multi-focus) plenoptic camera model

Linking a scene point $\boldsymbol{p}_w = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^\top$ to a blur aware plenoptic (BAP) feature $\boldsymbol{p} = \begin{bmatrix} u & v & \rho & 1 \end{bmatrix}^\top$ through each micro-lens (k, l) of type (i) [2]:

$$
\begin{aligned}
\textbf{Direct projection:} \quad & \begin{bmatrix} u & v & \rho & 1 \end{bmatrix}^\top \propto \Pi_{k,l}\left(\begin{bmatrix} x & y & z & 1 \end{bmatrix}^\top\right) \\
\textbf{Inverse projection:} \quad & \begin{bmatrix} x & y & z & 1 \end{bmatrix}^\top \propto \Pi_{k,l}^{-1}\left(\begin{bmatrix} u & v & \rho & 1 \end{bmatrix}^\top\right)
\end{aligned}
\tag{2}
$$

[2] Labussière et al., "Leveraging blur information for plenoptic camera calibration" (*IJCV, 2022*)

(Multi-focus) plenoptic camera model



Background

Blur model

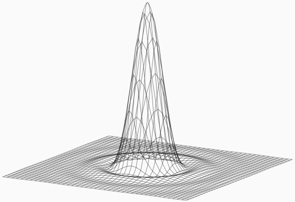
Blur modeling

Geometric blur $\longrightarrow$ circle of confusion, characterized by its blur radius ρ

Blur modeling

Geometric blur $\longrightarrow$ circle of confusion, characterized by its blur radius ρ

Physical blur $\longrightarrow$ point-spread function $\mathrm{h}(x, y)$, characterized by its spread parameter σ



Blur modeling

Geometric blur $\longrightarrow$ circle of confusion, characterized by its blur radius ρ

Physical blur $\longrightarrow$ point-spread function $\mathrm{h}(x, y)$, characterized by its spread parameter σ

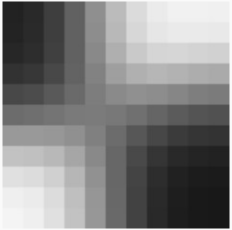
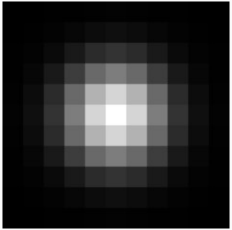
$$\sigma = \kappa \cdot \rho \tag{3}$$

where κ is a camera constant determined by calibration [2].

[2] Labussière et al., "Leveraging blur information for plenoptic camera calibration" (*IJCV, 2022*)

Relative blur

For 2 different micro-lens types (i) and (j), the equally-defocused representation is

 $=$  $*$ 

$$\mathcal{I}_{(i)}(x, y) \quad = \quad \mathrm{h}_r \quad * \quad \mathcal{I}_{(j)}(x, y)$$

Relative blur as function of the virtual depth

The relative blur can be approximated by a linear function of the disparity, i.e., of the inverse virtual depth up to a factor, such that

$$\rho_r^2 = \rho_{(i)}^2 - \rho_{(j)}^2 \approx m_{i,j} \cdot \upsilon^{-1} + q_{i,j} \tag{4}$$

where $m_{i,j}$ and $q_{i,j}$ are known thanks to the camera calibration.

Method

Contributions

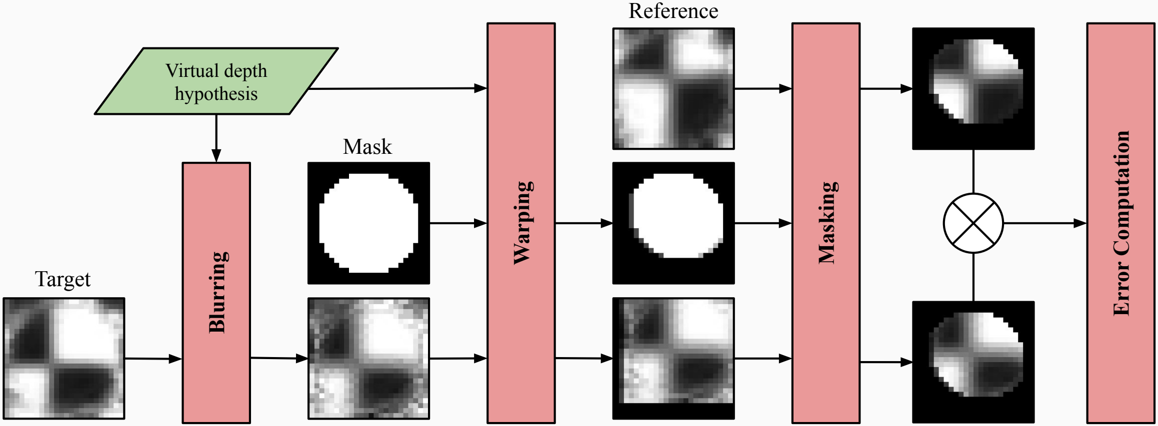
Our contributions are two-folds:

- introduce a metric depth estimation framework for plenoptic cameras working directly on raw images
- present a new dataset of 3D real-world scenes with GTs acquired with a 3D lidar scanner

---

This is achieved by introducing a new blur aware similarity error formulation leveraging both spatially-variant blur and disparity cues between micro-images

Blur aware similarity error formulation



Blur aware depth estimation (BLADE)

We estimate a raw depth map $\mathcal{D}$ directly from raw plenoptic images.

Two variations are considered:

Blur aware depth estimation (BLADE)

We estimate a raw depth map $\mathcal{D}$ directly from raw plenoptic images.

Two variations are considered:

1. coarse estimation, i.e., one depth per micro-image; **pros:** faster

Blur aware depth estimation (BLADE)

We estimate a raw depth map $\mathcal{D}$ directly from raw plenoptic images.

Two variations are considered:

1. coarse estimation, i.e., one depth per micro-image; **pros:** faster
2. refined estimation, i.e., one depth per pixel **pros:** more details

Blur aware depth estimation

Blur aware depth estimation

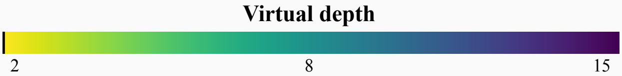


Figure 2: Coarse depth map

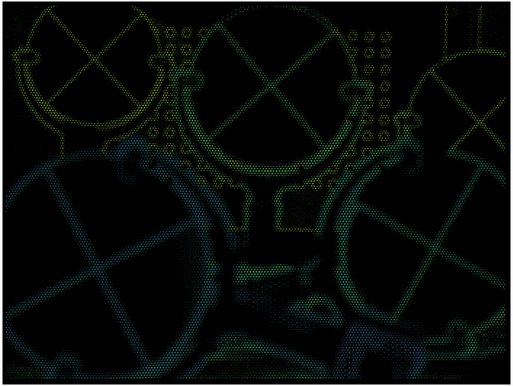
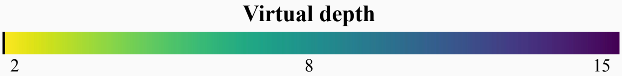


Figure 3: Refined depth map

Blur aware depth estimation



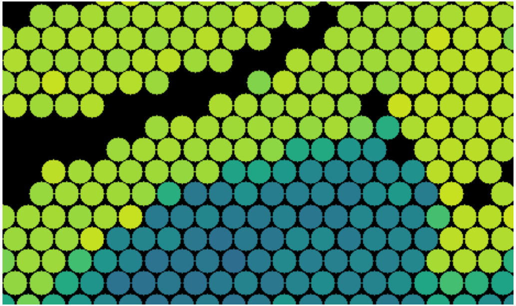


Figure 4: Zoom on coarse depth map

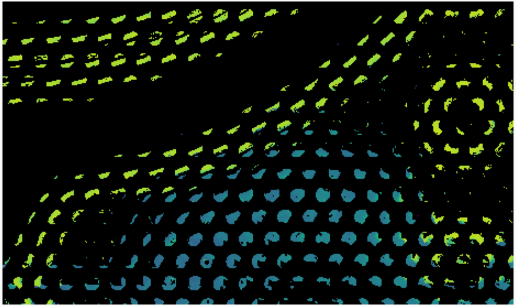


Figure 5: Zoom on refined depth map

Depth map computation

Input: Raw image, Camera model

Depth map computation

Input: Raw image, Camera model
Output: Coarse depth map $\mathcal{D}(k, l)$

Depth map computation

Input: Raw image, Camera model

Output: Coarse depth map $\mathcal{D}(k, l)$

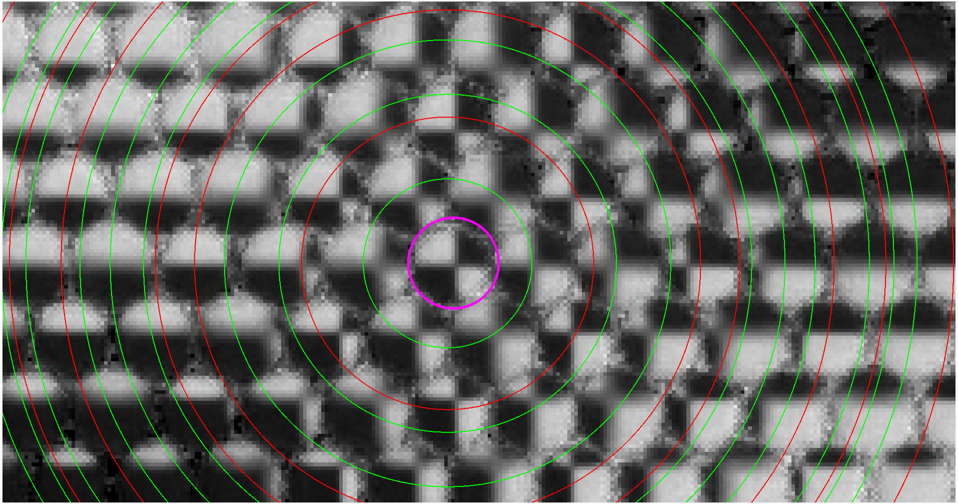
1: **for all** micro-image $\mathcal{I}$ with enough texture **do**

Depth map computation

Input: Raw image, Camera model
Output: Coarse depth map $\mathcal{D}(k, l)$

1: **for all** micro-image $\mathcal{I}$ with enough texture **do**
2: retrieve default neighborhood $\mathcal{N}(\mathcal{I})$

Depth map computation



Depth map computation

Input: Raw image, Camera model
Output: Coarse depth map $\mathcal{D}(k, l)$

1: **for all** micro-image $\mathcal{I}$ with enough texture **do**
2: retrieve default neighborhood $\mathcal{N}(\mathcal{I})$
3: compute initial virtual depth v_0
4: $\mathcal{D}(k, l) \leftarrow v_0$

Depth map computation

Input: Raw image, Camera model
Output: Coarse depth map $\mathcal{D}(k,l)$

1: **for all** micro-image $\mathcal{I}$ with enough texture **do**
2: retrieve default neighborhood $\mathcal{N}(\mathcal{I})$
3: compute initial virtual depth v_0
4: $\mathcal{D}(k,l) \leftarrow v_0$
5: update neighborhood $\mathcal{N}(\mathcal{I}, v_0)$

Depth map computation

Input: Raw image, Camera model
Output: Coarse depth map $\mathcal{D}(k,l)$

for all micro-image $\mathcal{I}$ with enough texture **do**
    retrieve default neighborhood $\mathcal{N}(\mathcal{I})$
    compute initial virtual depth v_0
    $\mathcal{D}(k,l) \leftarrow v_0$
    update neighborhood $\mathcal{N}(\mathcal{I}, v_0)$
    compute virtual depth $\hat{v}$
    $\mathcal{D}(k,l) \leftarrow \hat{v}$
end for

Depth map computation

Input: Raw image, Camera model
Output: Coarse depth map $\mathcal{D}(k,l)$

```
for all micro-image 𝓘 with enough texture do
    retrieve default neighborhood 𝓝(𝓘)
    compute initial virtual depth v₀
    𝓓(k,l) ← v₀
    update neighborhood 𝓝(𝓘, v₀)
    compute virtual depth v̂
    𝓓(k,l) ← v̂
end for
convert virtual to metric using Π⁻¹_{k,l}
```

Experiment

1. Background
2. Method
3. Experiment
4. Conclusion

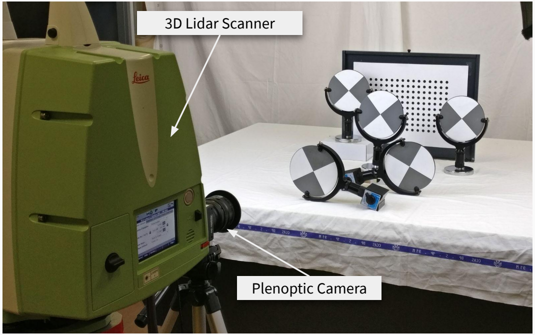
Absolute depth experimental setup (R12-E, ES, ELP20)

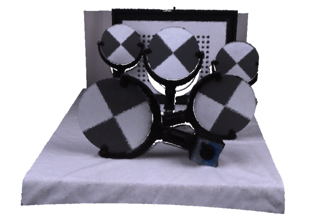
`Raytrix R12`, multi-focus plenoptic camera with

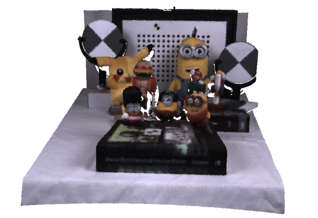
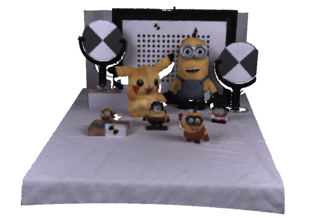
- mounted lens of 50mm focal length,
- focus distance $h = 2133$ mm.

3D scenes acquired with a 3D lidar scanner, a `Leica ScanStation P20`

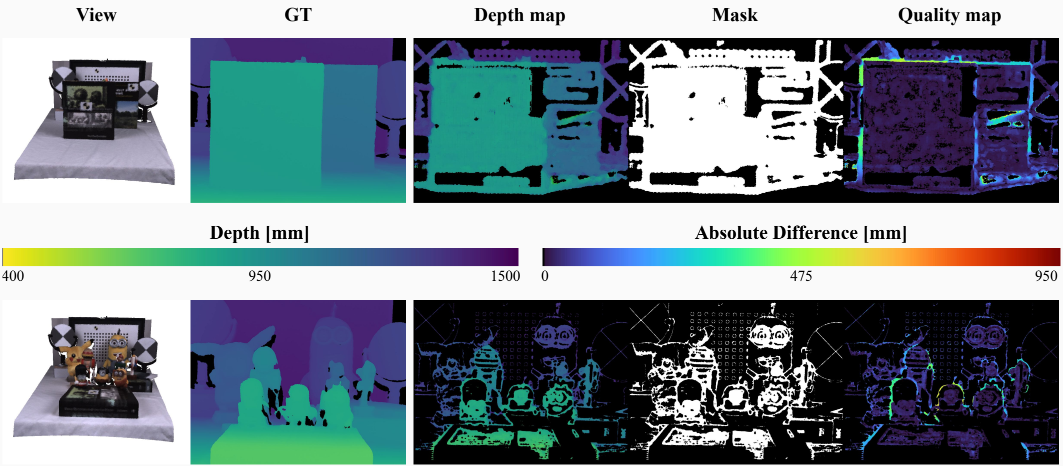
- calibration scene with target,
- easy scenes with planar objects,
- complex scenes with figurines.



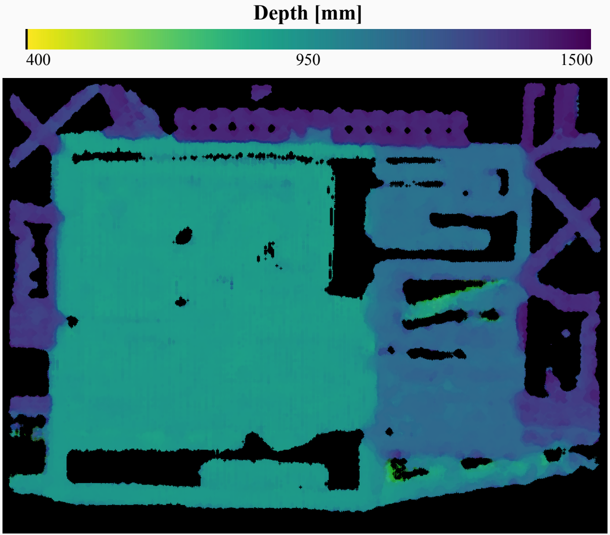


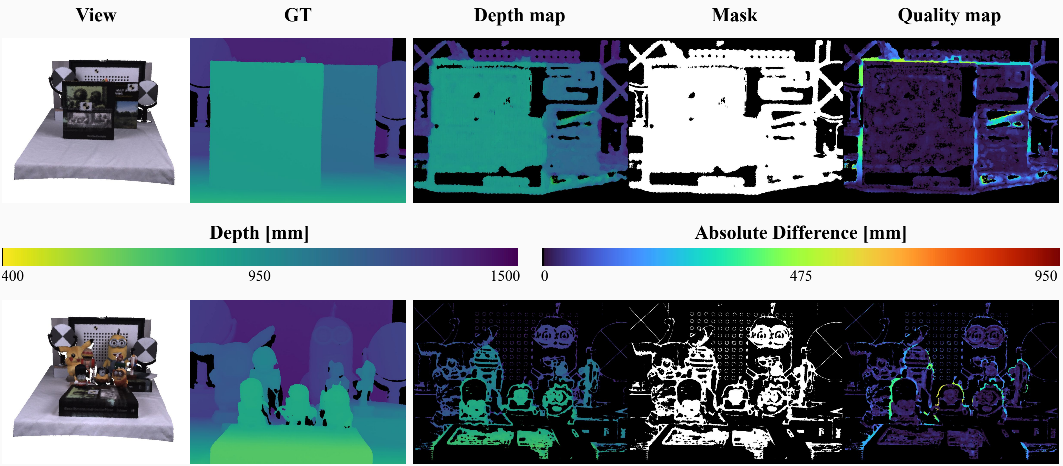
Absolute depth evaluation



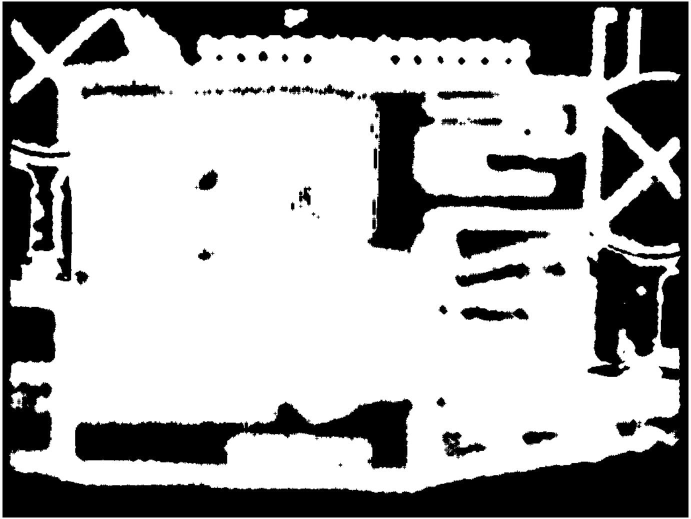
Absolute depth evaluation



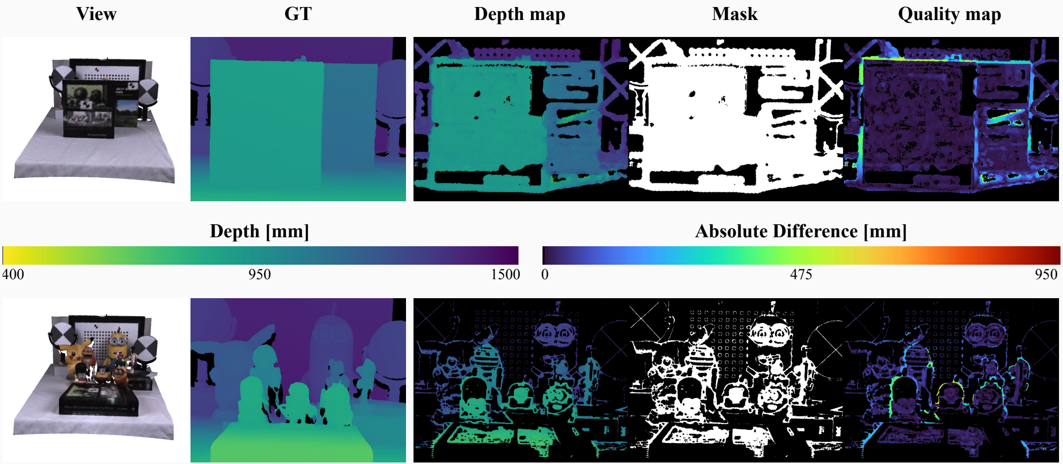
Absolute depth evaluation



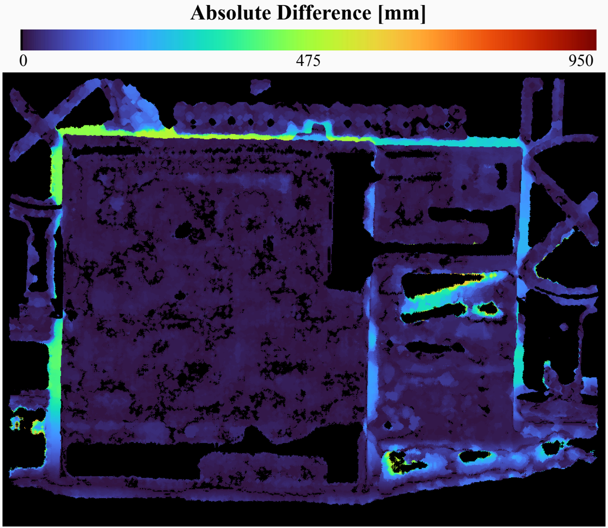
Absolute depth evaluation



Absolute depth evaluation



Absolute depth evaluation



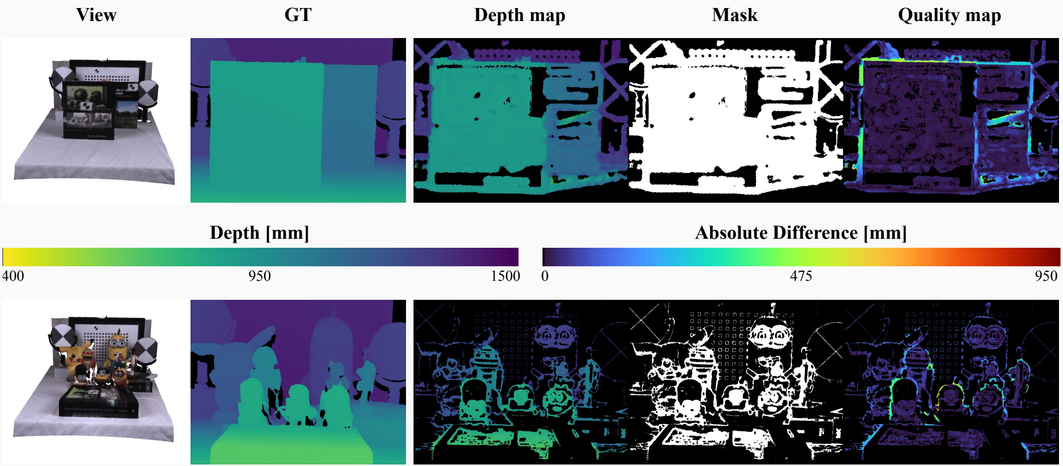
Errors:

$\mathcal{Q}_{25} = 5.482$ mm

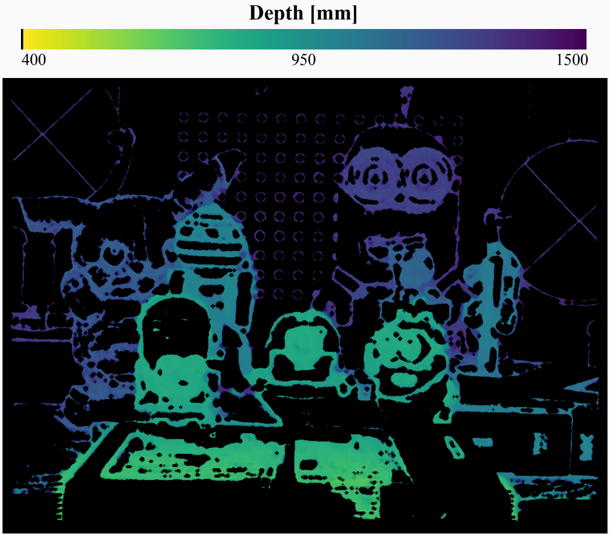
median $= 11.370$ mm

$\mathcal{Q}_{75} = 24.468$ mm

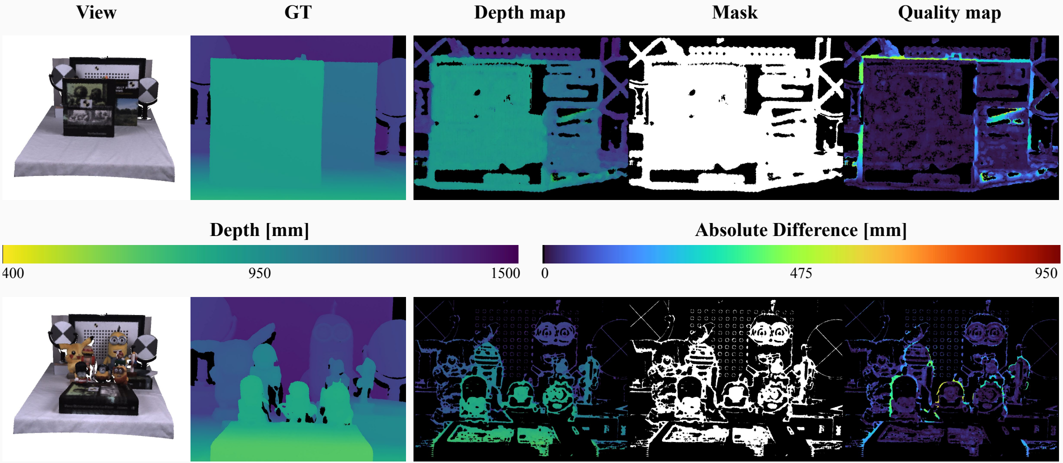
Absolute depth evaluation



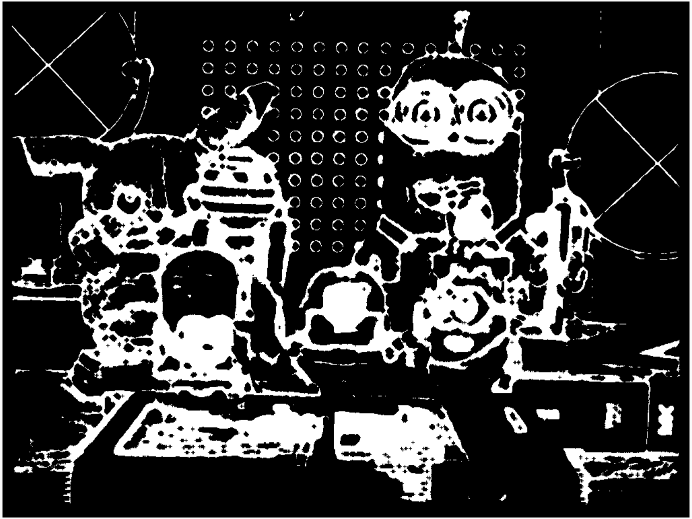
Absolute depth evaluation



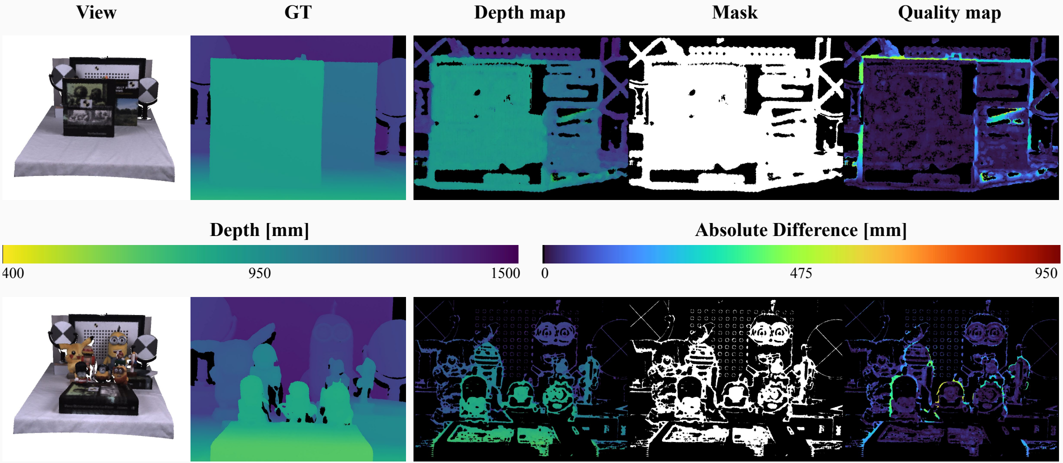
Absolute depth evaluation



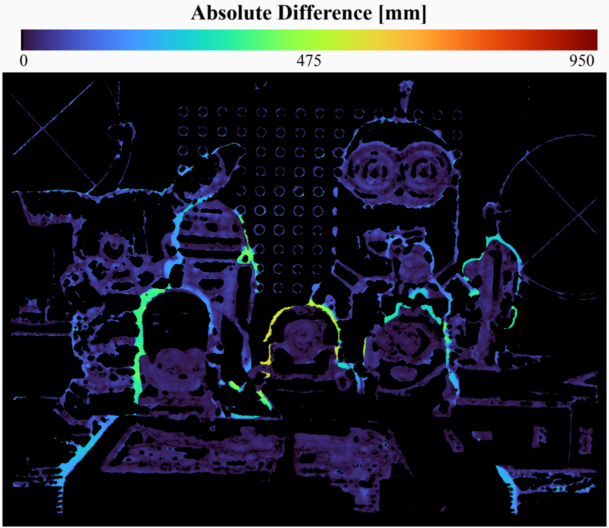
Absolute depth evaluation



Absolute depth evaluation



Absolute depth evaluation



Errors:

$\mathcal{Q}_{25} = 12.011$ mm

median $= 25.282$ mm

$\mathcal{Q}_{75} = 50.377$ mm

Absolute depth evaluation

Leveraging blur in our coarse depth estimation framework:
$\longrightarrow$ mean median-error of 19 mm, for distances ranging from 400 mm to 1500 mm.

Relative errors in function of the distances are

- for all scenes, from 1.27 % to 4.75 %,
- for easy scenes, from 0.96 % to 3.59 %,
- for complex scenes, from 1.73 % to 6.49 %.

Conclusion

Concluding remarks on depth estimation

Leveraging blur information we proposed:

- a depth estimation framework, operating in raw image space and taking advantages of both defocus and correspondence cues

  $\longrightarrow$ introducing defocus cues improves the depth estimation

  $\longrightarrow$ accurate and precise metric depth estimates

Validated by quantitative evaluations on real-world 3D complex scenes.

Improvements

- Blur model $\longrightarrow$ evaluate other PSF models and calibrate for RGB

Improvements

- Blur model $\longrightarrow$ evaluate other PSF models and calibrate for RGB
- Depth estimation

  $\longrightarrow$ use RGB image

  $\longrightarrow$ implementation on GPU

  $\longrightarrow$ refinement and global optimization

Perspectives

- Application to metrology of rain droplets

  $\longrightarrow$ characterize the rain profile, i.e., droplet size, quantity, velocity, etc.

  $\longrightarrow$ improve depth estimation by taking into account the droplet occlusions

Perspectives

- Application to metrology of rain droplets
  - $\longrightarrow$ characterize the rain profile, i.e., droplet size, quantity, velocity, etc.
  - $\longrightarrow$ improve depth estimation by taking into account the droplet occlusions
- Application to robotics
  - $\longrightarrow$ visual servoing, 3D reconstruction, localization and mapping

Perspectives

- Application to metrology of rain droplets
  - $\longrightarrow$ characterize the rain profile, i.e., droplet size, quantity, velocity, etc.
  - $\longrightarrow$ improve depth estimation by taking into account the droplet occlusions
- Application to robotics
  - $\longrightarrow$ visual servoing, 3D reconstruction, localization and mapping
- Application to co-design of plenoptic camera and depth estimation
  - $\longrightarrow$ design simultaneously the optics parameters and the desired application

Thanks! Any Questions?

References i

- Mathieu Labussière et al. "Blur Aware Calibration of Multi-Focus Plenoptic Camera". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. **Oral**: IEEE, June CVPR, 2020, pp. 2542–2551.
- Mathieu Labussière et al. "Leveraging blur information for plenoptic camera calibration". In: *International Journal of Computer Vision* (IJCV, 2022), pp. 1–22.
- Charles Antoine Noury, Céline Teulière, and Michel Dhome. "Light-Field Camera Calibration from Raw Images". In: *DICTA 2017 – International Conference on Digital Image Computing: Techniques and Applications* (2017), pp. 1–8.
- Christian Heinze et al. "Automated Robust Metric Calibration Algorithm for Multifocus Plenoptic Cameras". In: *IEEE Transactions on Instrumentation and Measurement* 65.5 (2016), pp. 1197–1205.

Relative depth experimental setup (R12-ABC)

`Raytrix R12`, multi-focus plenoptic camera with:

- mounted lens of **50mm** focal length,
- a MLA composed of 3 micro-lens types,
- a pixel size of $s = 5.5$µm.

Building 3 datasets for 3 focus distances h:

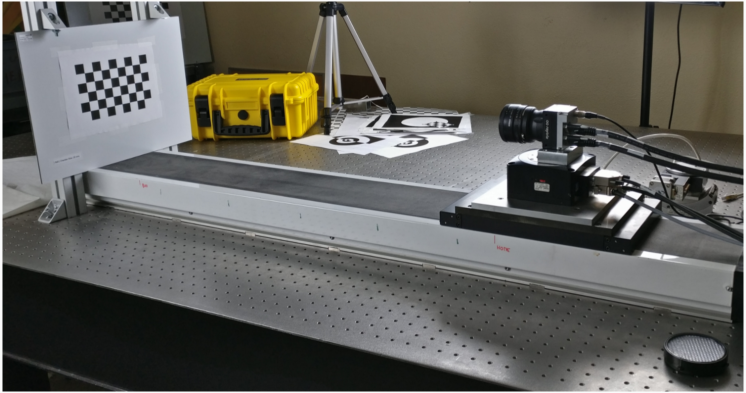
- R12-A for $h = 450$ mm,
- R12-B for $h = 1000$ mm,
- R12-C for $h = \infty$.

Relative depth error evaluation

Comparison with state-of-the-art methods and without scale correction

$\longrightarrow$ disparity only with intrinsics of Noury et al. (*2017*) [3]

$\longrightarrow$ from the proprietary software RxLive, Heinze et al. (*2016*) [4]



Relative depth error evaluation

Relative depth error w.r.t. the ground truth displacement