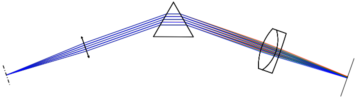
Differentiable ray-tracer used for coded hyperspectral systems co-design



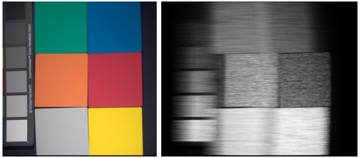
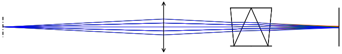
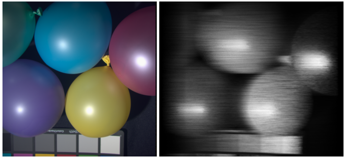


PHOTO LAAS-CNRS, SISU IRAP

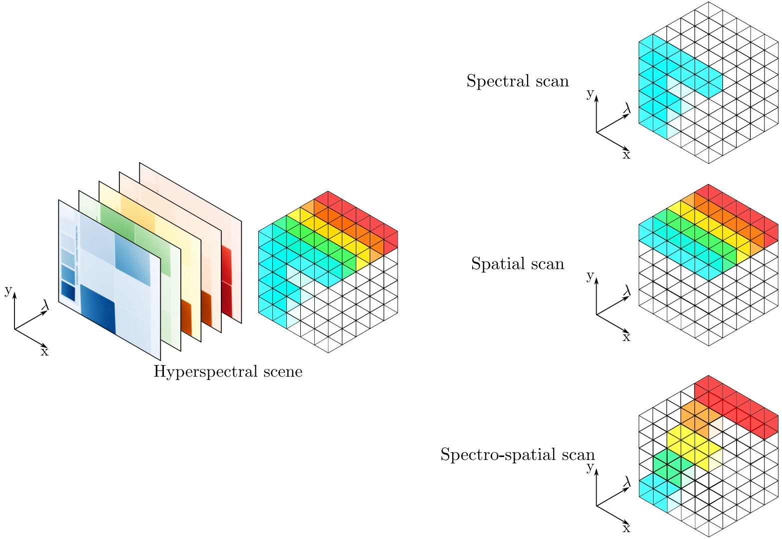




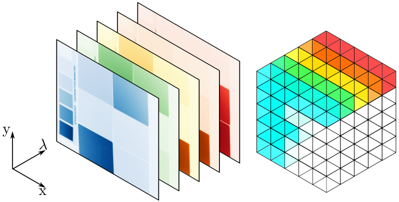
Léo Paillet, Hervé Carfantan, Simon Lacroix and Antoine Monmayrant

Context

Classical hyperspectral imaging



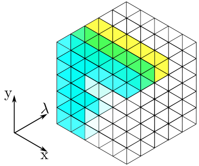
Classical hyperspectral imaging



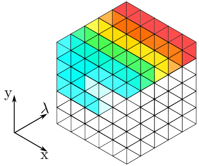
Hyperspectral scene

- → High data volume
- → High acquisition time / noise
- → Redundant information

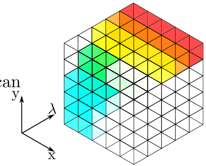
Spectral scan



Spatial scan



Spectro-spatial scan

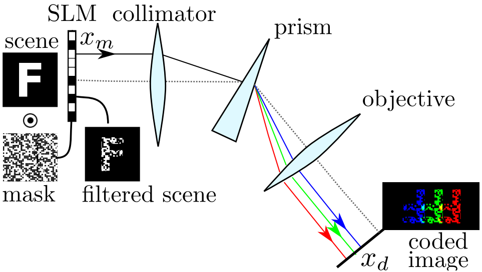


Principles

- Inspired by compressed sensing
- Non-traditional method : CASSI
- Based on a coded aperture named «mask» (blocks certain rays)

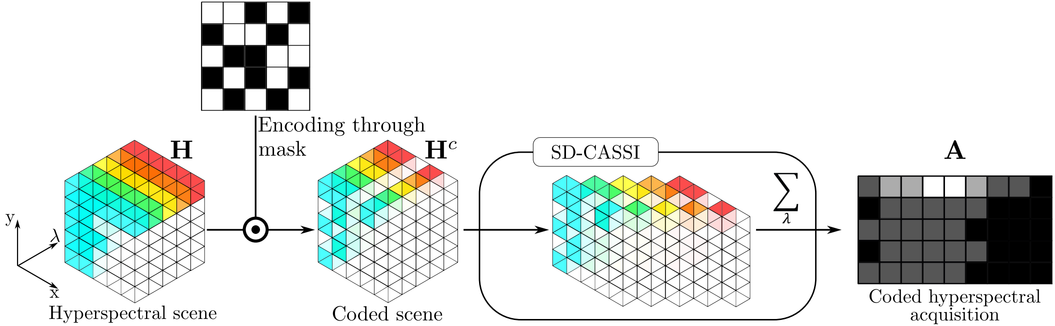
Features

- No spatial/spectral resolution sacrifice
- Spatio-spectral mix
- Less acquisitions, noise



SD-CASSI : Single Disperser Coded Aperture Snapshot Spectral Imager

Hyperspectral coded (HSC) imaging with SD-CASSI



Simple model SD-CASSI : Single Disperser Coded Aperture Snapshot Spectral Imager

- Simple model : $\mathbf{A}(x,y) = \sum_\lambda \mathbf{H}^c(x - s(\lambda), y, \lambda)$ with often $s(\lambda) = k\lambda$ and no impulse response
- Acquisitions depend on optics and mask parameters

[Wagadarikar et al., 2008, 10.1364/AO.47.000B44]

CASSI Differentiable Simulator

Our simulator : DiffCassiSim

- Accurate modelling of any optics based on dO (github DiffOptics)
  - Differentiable (PyTorch), fast, RAM-light ray tracing ($\sim$several millions), made for lenses
  - Images rendering
- Modifications
  - Added prisms, mirrors, thin lenses, can import .zmx models

[Wang et al., 2022, 10.1109/TCI.2022.3212837]

Our simulator : DiffCassiSim

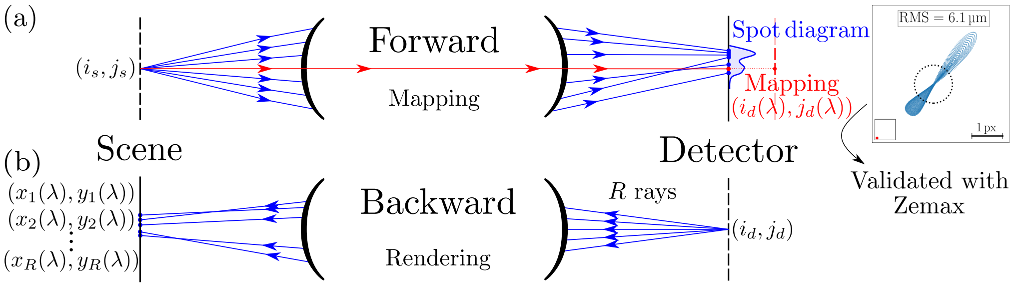
- Accurate modelling of any optics based on dO (github DiffOptics)
  - Differentiable (PyTorch), fast, RAM-light ray tracing ($\sim$several millions), made for lenses
  - Images rendering
- Modifications
  - Added prisms, mirrors, thin lenses, can import .zmx models
  - Axial symmetry break thanks to rotations and translations between elements

[Wang et al., 2022, 10.1109/TCI.2022.3212837]

- Accurate modelling of any optics based on dO (github DiffOptics)
  - Differentiable (PyTorch), fast, RAM-light ray tracing ($\sim$several millions), made for lenses
  - Images rendering
- Modifications
  - Added prisms, mirrors, thin lenses, can import .zmx models
  - Axial symmetry break thanks to rotations and translations between elements
- CASSI systems co-design
- Usable for other systems than CASSI

[Wang et al., 2022, 10.1109/TCI.2022.3212837]

Forward and backward modes



- Mapping : forward : from scene to detector
  - Describes image formation model
- Rendering : backward : from detector to scene (classical method)

Image formation model

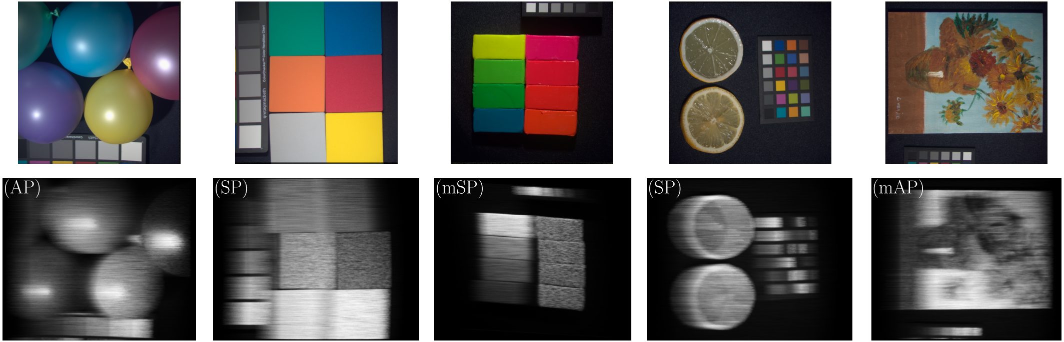
Acquisition $\mathbf{A}$: $\mathbf{A}(i_d, j_d) = \sum_{r=1}^{R} \mathbf{H}^c(x_r(k), y_r(k), k)$ with $\mathbf{H}^c$ the coded scene

Acquisition $\mathbf{A}$: $\mathbf{A}(i_d, j_d) = \sum_{r=1}^{R} \underbrace{\mathbf{H}^c(x_r(k), y_r(k), k)}_{\text{approximated through interpolation}}$

with $\mathbf{H}^c$ the coded scene

$\Longrightarrow$ Accounts for any data arrangement

Examples of simulated acquisitions

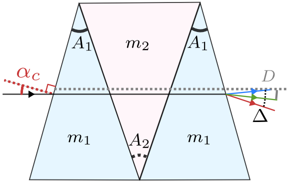


Dataset : CAVE/KAIST

Simulations to evaluate CASSI designs : Unrolling learning

What is a good CASSI system ?

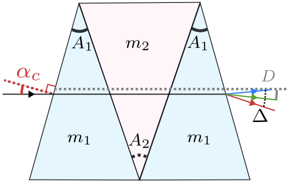
- Use of AMICI prisms to reduce distortions



[Paillet et al., 2025]

What is a good CASSI system ?

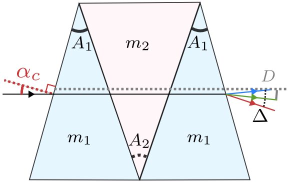
- Use of AMICI prisms to reduce distortions
  - Is it necessary and optimal ?



[Paillet et al., 2025]

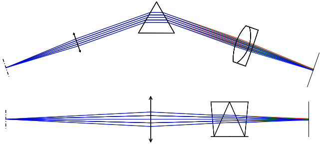
What is a good CASSI system ?

- Use of AMICI prisms to reduce distortions
  - Is it necessary and optimal ?



The question : Do distortions matter ?

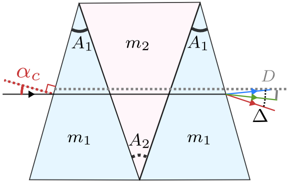
- Compare different CASSI systems
  - Different distortions
  - Different alignments
  - Similar dispersions



[Paillet et al., 2025]

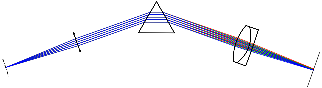
What is a good CASSI system ?

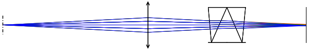
- Use of AMICI prisms to reduce distortions
  - Is it necessary and optimal ?



The question : Do distortions matter ?

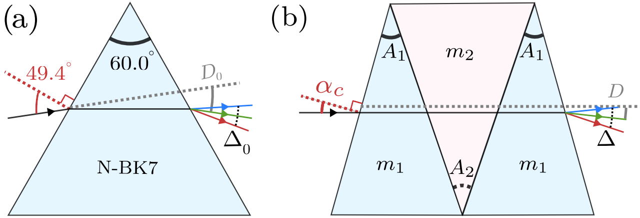
- Compare different CASSI systems
  - Different distortions
  - Different alignments
  - Similar dispersions
- Task : reconstruction
- SoTA Unrolling Algorithm accounting for the model





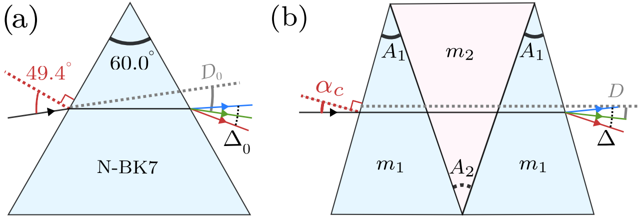
[Paillet et al., 2025]

Two comparable prisms



- Single prism : smile distortions | Amici prism : negligible distortions

Two comparable prisms



- Single prism : smile distortions | Amici prism : negligible distortions
- Optimization of Amici prism to be comparable to single prism
  - Same dispersion ($\Delta \simeq \Delta_0$), optical properties
  - Small distortions (~30 times lower), no deviation, compact

[Wagadarikar et al., 2008, 10.1364/AO.47.000B44 ; Wagadarikar et al., 2009, 10.1364/OE.17.006368]

[Wang et al., 2015, 10.1109/CVPR.2015.7299128]

Goal

- Optimization of materials and apex angles

Prism optimization

Goal

- Optimization of materials and apex angles

Process : Trace rays from scene to detector to compute losses

- Differentiable materials optimization : Continuous (n_d, V_d) for Cauchy's law
  - Validity loss by comparing to catalog (SCHOTT, CDGM, ...) n_d's and V_d's

Goal

- Optimization of materials and apex angles

Process : Trace rays from scene to detector to compute losses

- Differentiable materials optimization : Continuous (n_d, V_d) for Cauchy's law
  - Validity loss by comparing to catalog (SCHOTT, CDGM, ...) n_d's and V_d's
- From grid of points for distortions
- From center of FOV for dispersions and deviation
  - Comparison of dispersion to reference single prism

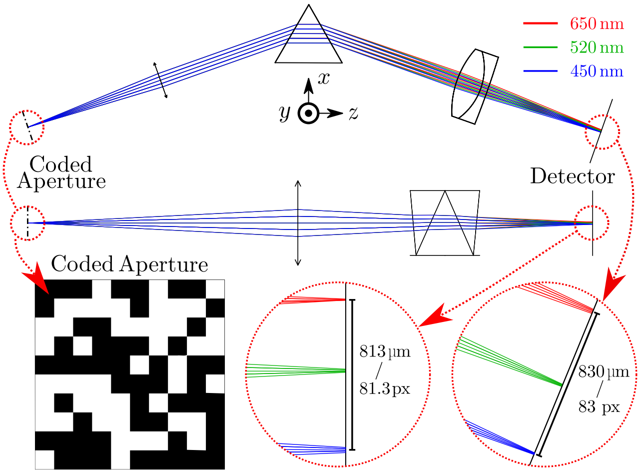
Goal

- Optimization of materials and apex angles

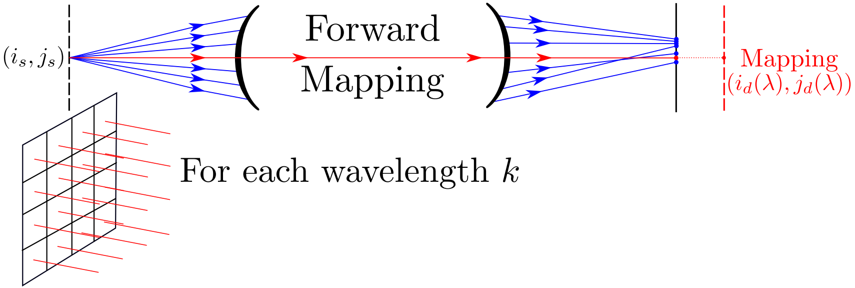
Process : Trace rays from scene to detector to compute losses

- Differentiable materials optimization : Continuous (n_d, V_d) for Cauchy's law
  - Validity loss by comparing to catalog (SCHOTT, CDGM, ...) n_d's and V_d's
- From grid of points for distortions
- From center of FOV for dispersions and deviation
  - Comparison of dispersion to reference single prism
- Second round : fixed, existing materials
  - Focus only on angles optimization

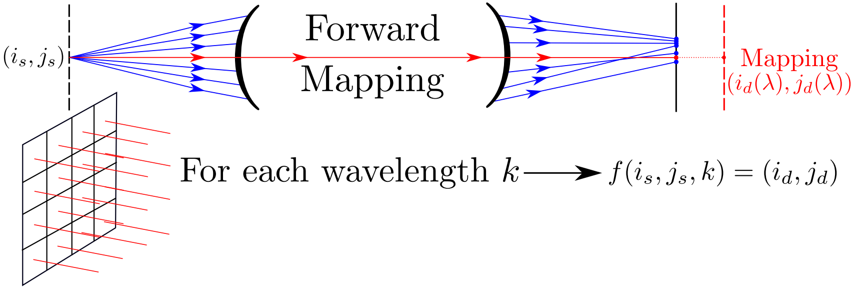
Modelled systems with DiffCassiSim



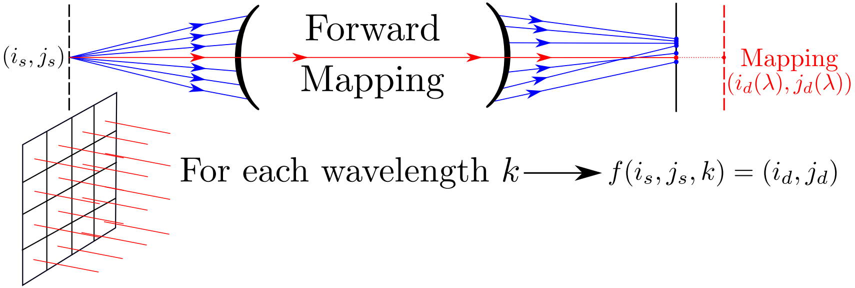
Mapping function to account for optical model



Mapping function to account for optical model

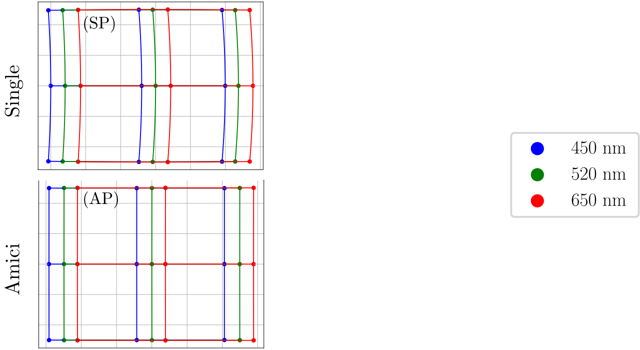


Mapping function to account for optical model

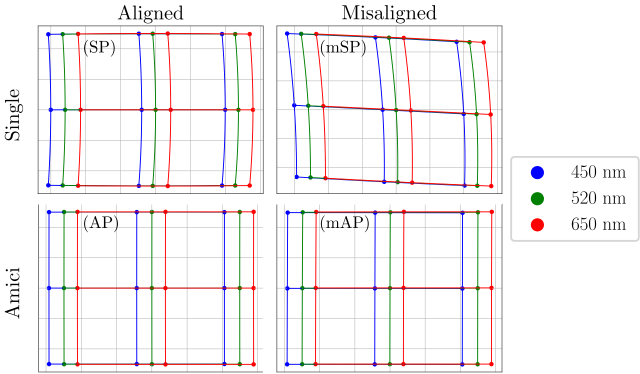


- Exploit the optical model during reconstruction
- Represents adjoint operator in the algorithm $\Longrightarrow$ Ensure co-design accounting for the image formation model

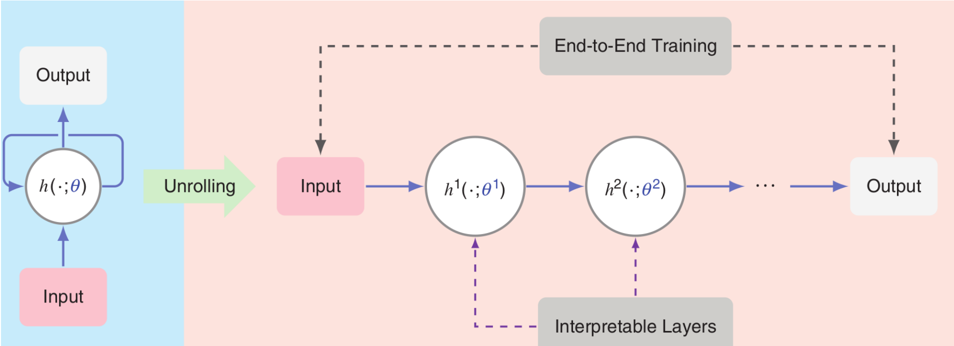
Distortions



Distortions



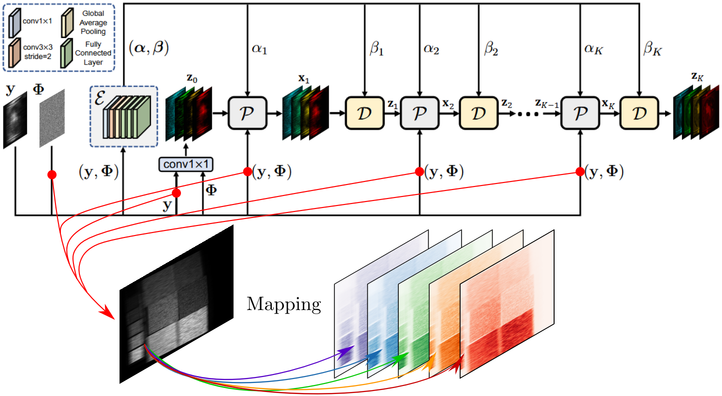
Unrolling algorithms



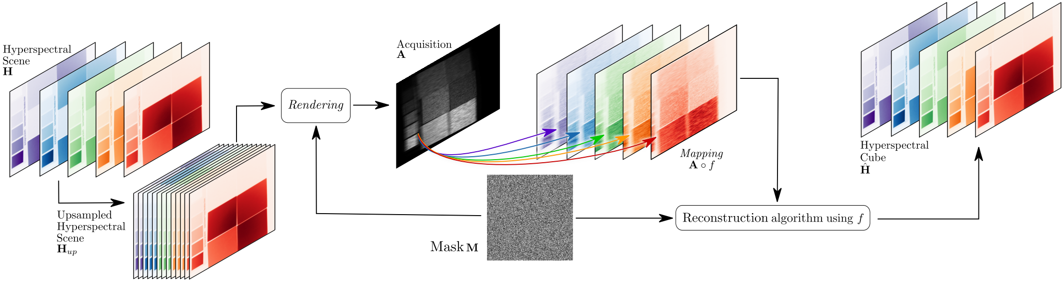
[Monga et al., 2021, 10.1109/msp.2020.3016905]

- Learns iterative algorithm parameters
- More interpretable

Unrolling hyperspectral reconstruction using optical model



Algorithm : DAUHST[Cai et al., 2022] (DGSMP, MST, PADUT, RDLUF)



- Same mask during all training : opening ratio (ROM) of 50%

Results with DGSMP, MST, DAUHST, RDLUF, PADUT

		(AP)	(SP)
RMSE ↓ ($\times 10^{-3}$)	DGSMP	**31.3**	32.8
	MST	27.3	**26.5**
	DAUHST	20.7	19.6
	RDLUF	**22.6**	24.1
	PADUT	21.4	**19.6**
PSNR ↑	DGSMP	**30.6**	30.3
	MST	31.9	**32.2**
	DAUHST	34.4	34.7
	RDLUF	**33.5**	32.9
	PADUT	34.1	**34.7**
SSIM ↑ $[0-1]$	DGSMP	**0.892**	0.883
	MST	0.910	**0.914**
	DAUHST	0.942	**0.945**
	RDLUF	**0.930**	0.922
	PADUT	0.937	**0.943**
SAM ↓ $[0-1]$	DGSMP	**0.058**	0.062
	MST	0.055	**0.052**
	DAUHST	0.048	0.048
	RDLUF	**0.053**	0.055
	PADUT	**0.048**	**0.048**

- Similar performances across all four configurations
- No best configuration
- Each configuration reaches the best metric at least once

[Huang et al., 2021, 10.1109/CVPR46437.2021.01595 ; Cai et al., 2022, 10.1109/CVPR52688.2022.01698]

[Cai et al., 2022 ; Dong et al., 2023, 10.1109/CVPR52729.2023.02132 ; Li et al., 2023, 10.1109/ICCV51070.2023.01191]

Results with DGSMP, MST, DAUHST, RDLUF, PADUT

		(AP)	(SP)	(mAP)	(mSP)
RMSE ↓ ($\times 10^{-3}$)	DGSMP	**31.3**	32.8	31.8	34.4
	MST	27.3	**26.5**	**26.5**	27.0
	DAUHST	20.7	19.6	**19.5**	20.3
	RDLUF	**22.6**	24.1	23.8	23.7
	PADUT	21.4	**19.6**	21.6	21.8
PSNR ↑	DGSMP	**30.6**	30.3	30.5	29.9
	MST	31.9	**32.2**	32.1	32.0
	DAUHST	34.4	34.7	**34.9**	34.4
	RDLUF	**33.5**	32.9	33.0	33.0
	PADUT	34.1	**34.7**	34.0	33.8
SSIM ↑ [0 – 1]	DGSMP	**0.892**	0.883	0.886	0.879
	MST	0.910	**0.914**	0.912	0.911
	DAUHST	0.942	**0.945**	**0.945**	0.939
	RDLUF	**0.930**	0.922	0.926	0.926
	PADUT	0.937	**0.943**	0.936	0.935
SAM ↓ [0 – 1]	DGSMP	**0.058**	0.062	0.059	0.065
	MST	0.055	**0.052**	0.055	0.055
	DAUHST	0.048	0.048	**0.047**	0.050
	RDLUF	**0.053**	0.055	0.054	**0.053**
	PADUT	**0.048**	**0.048**	0.049	**0.048**

- Similar performances across all four configurations
- No best configuration
- Each configuration reaches the best metric at least once

[Huang et al., 2021, 10.1109/CVPR46437.2021.01595 ; Cai et al., 2022, 10.1109/CVPR52688.2022.01698]

[Cai et al., 2022 ; Dong et al., 2023, 10.1109/CVPR52729.2023.02132 ; Li et al., 2023, 10.1109/ICCV51070.2023.01191]

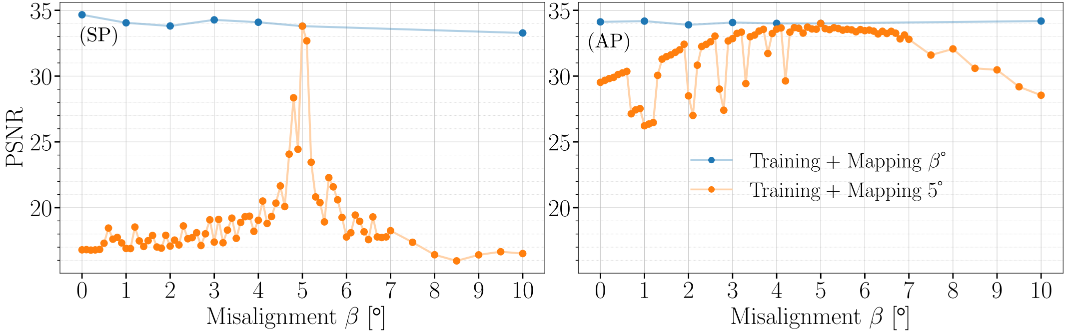
Results with DGSMP, MST, DAUHST, RDLUF, PADUT

		(AP)	(SP)	(mAP)	(mSP)
RMSE ↓ ($\times10^{-3}$)	DGSMP	**31.3**	32.8	31.8	34.4
	MST	27.3	**26.5**	**26.5**	27.0
	DAUHST	20.7	19.6	**19.5**	20.3
	RDLUF	**22.6**	24.1	23.8	23.7
	PADUT	21.4	**19.6**	21.6	21.8
PSNR ↑	DGSMP	**30.6**	30.3	30.5	29.9
	MST	31.9	**32.2**	32.1	32.0
	DAUHST	34.4	34.7	**34.9**	34.4
	RDLUF	**33.5**	32.9	33.0	33.0
	PADUT	34.1	**34.7**	34.0	33.8
SSIM ↑ $[0-1]$	DGSMP	**0.892**	0.883	0.886	0.879
	MST	0.910	**0.914**	0.912	0.911
	DAUHST	0.942	**0.945**	**0.945**	0.939
	RDLUF	**0.930**	0.922	0.926	0.926
	PADUT	0.937	**0.943**	0.936	0.935
SAM ↓ $[0-1]$	DGSMP	**0.058**	0.062	0.059	0.065
	MST	0.055	**0.052**	0.055	0.055
	DAUHST	0.048	0.048	**0.047**	0.050
	RDLUF	**0.053**	0.055	0.054	**0.053**
	PADUT	**0.048**	**0.048**	0.049	**0.048**

- Similar performances across all four configurations
- No best configuration
- Each configuration reaches the best metric at least once

[Huang et al., 2021, 10.1109/CVPR46437.2021.01595 ; Cai et al., 2022, 10.1109/CVPR52688.2022.01698]

[Cai et al., 2022 ; Dong et al., 2023, 10.1109/CVPR52729.2023.02132 ; Li et al., 2023, 10.1109/ICCV51070.2023.01191]


(SP)
(AP)
PSNR
35
30
25
20
0 1 2 3 4 5 6 7 8 9 10
Misalignment β [°]
Training + Mapping β°
Training + Mapping 5°

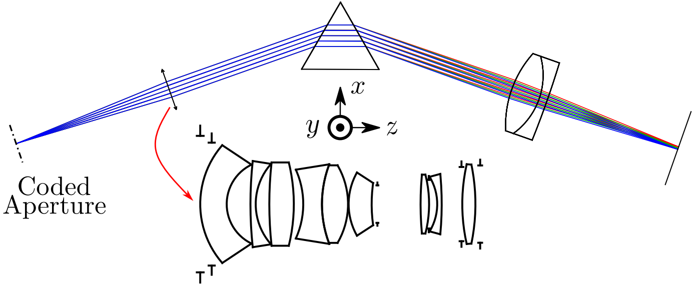
Conclusion

- Differentiable simulations are great to model physics while allowing for optimization and gradient flow
  - Useful for design, processing, co-design
- Best systems might not be what we think
- Distortions and misalignments have a marginal impact on processing as long as we know them and account for them

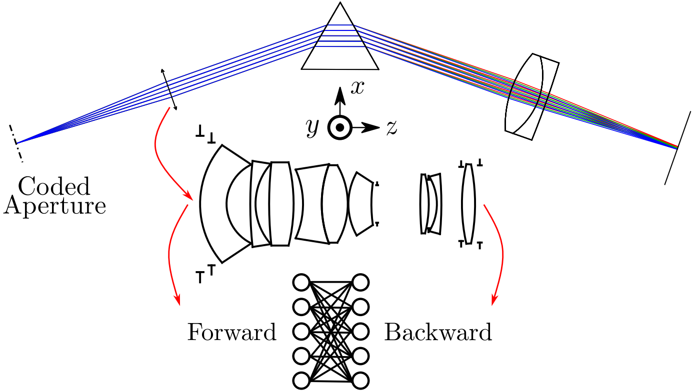
Limits and prospects

- No lens design (e.g. blackbox files) $\implies$ can't simulate $\implies$ proxy model
  - Hence the perfect lenses
- For complete co-design, optimizing from loss after processing is computationally heavy $\implies$ co-design from acquisitions

Proxy to unknown lens designs



Proxy to unknown lens designs



- Predict positions, angles, validity of rays
  - Currently $<$ 5µm error (14cm FOV), about 0.04° direction error, 99.8% correct classification, across 3 wavelengths
- Doesn't violate reversibility of light

Mask optimization

Reduce computational strain

- Optimization from a loss on the acquisitions

[Pinkard et al., 2025 ; Friedman et al., 2023]

Reduce computational strain

- Optimization from a loss on the acquisitions
  - Information gain

[Pinkard et al., 2025 ; Friedman et al., 2023]

Mask optimization

Reduce computational strain

- Optimization from a loss on the acquisitions
  - Information gain
  - Mask optimization without post-processing

[Pinkard et al., 2025 ; Friedman et al., 2023]

Mask optimization

Reduce computational strain

- Optimization from a loss on the acquisitions
    - Information gain
    - Mask optimization without post-processing
    - Behaves like losses on reconstruction (?)

[Pinkard et al., 2025 ; Friedman et al., 2023]

Mask optimization

Reduce computational strain

- Optimization from a loss on the acquisitions
  - Information gain
  - Mask optimization without post-processing
  - Behaves like losses on reconstruction (?)
- Mutual Information metric -> problems in high dimension

[Pinkard et al., 2025 ; Friedman et al., 2023]

Reduce computational strain

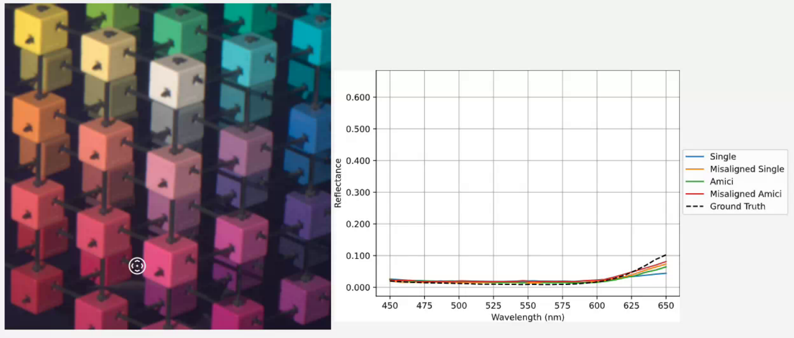
- Optimization from a loss on the acquisitions
  - Information gain
  - Mask optimization without post-processing
  - Behaves like losses on reconstruction (?)
- Mutual Information metric -> problems in high dimension
- Vendi Score -> samples only

[Pinkard et al., 2025 ; Friedman et al., 2023]

Thank you for your attention !

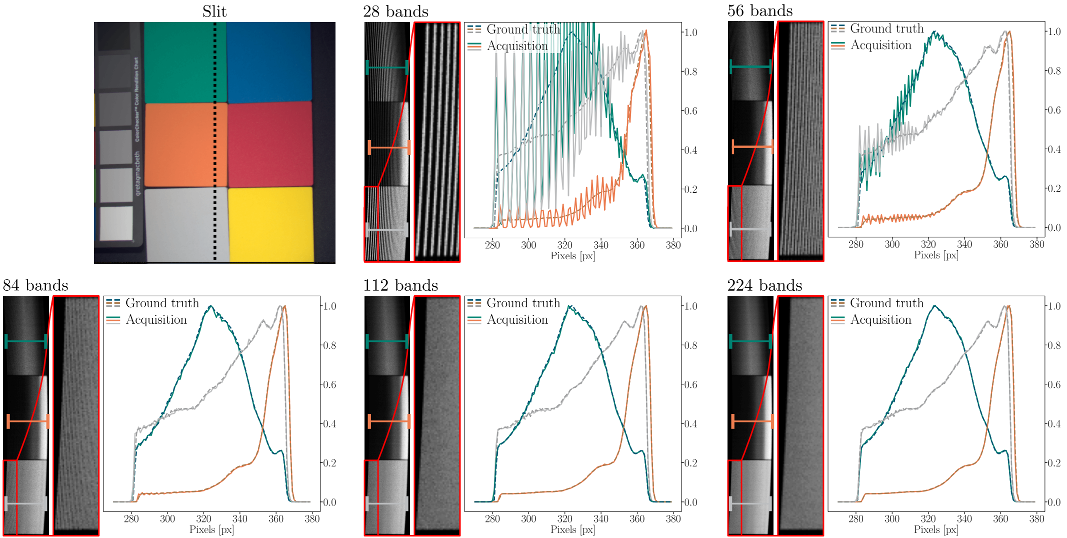
All the code is open source on github : DiffCassiSim

Annexe

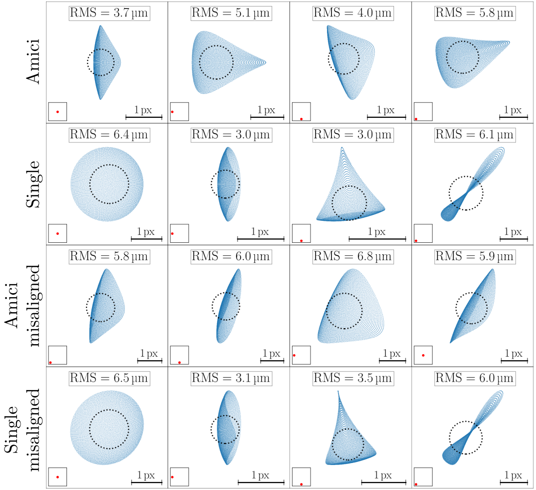


0.600
0.500
0.400
0.300
0.200
0.100
0.000
Reflectance
450
475
500
525
550
575
600
625
650
Wavelength (nm)
Single
Misaligned Single
Amici
Misaligned Amici
Ground Truth

Upsampling necessity



Spot diagrams at 650nm



Mutual Information

$$I(X,Y) = H(X) + H(Y) - H(X,Y) = H(Y) - H(Y|X)$$
$$H(Y) = -\sum_{y \in Y} p(y) \log_2 p(y)$$

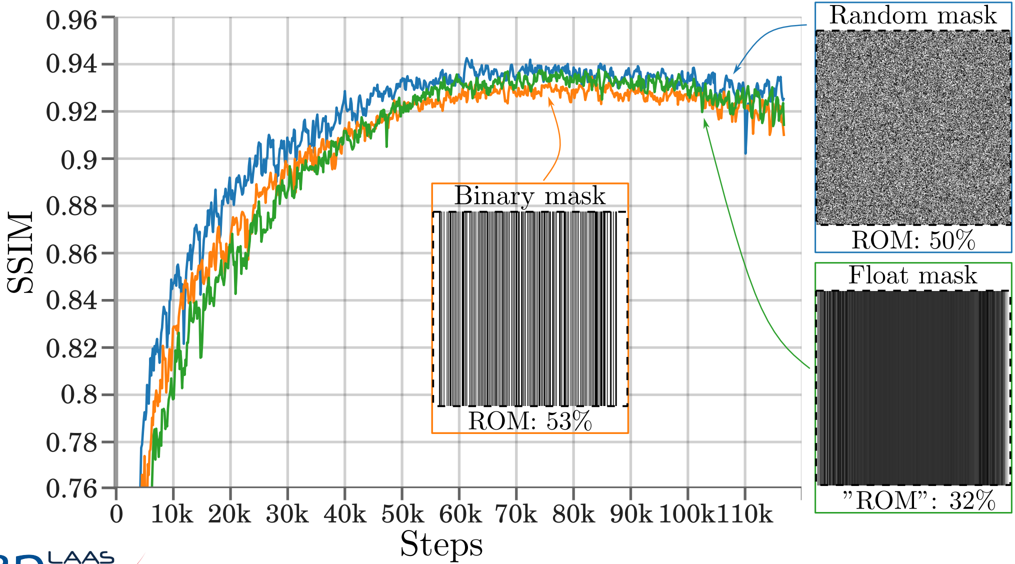
Vendi Score

- Based on 2 hyperparameters : a similarity kernel k and q
  - k such that $k(x,x) = 1$ and $k(x,y) \in [0,1]$ measures similarity between x and y

$$VS_q(D;q) = \exp\left(\frac{1}{1-q} \log\left(\sum_{i=1}^{n} (\bar{\lambda}_i)^q\right)\right)$$

with $D = \{x_1, ..., x_n\}$ dataset and $\bar{\lambda}_i$ eigenvalues of $K = (k(x_i, x_j))_{1 \leq i,j \leq n}$

Example optimization with Vendi Score



[1] A. Wagadarikar, R. John, R. Willett, D. Brady, *Appl. Opt.* 2008, *47*, B44, 10.1364/AO.47.000B44.

[2] C. Wang, N. Chen, W. Heidrich, *IEEE Transactions on Computational Imaging* 2022, *8*, 905, 10.1109/TCI.2022.3212837.

[3] L. Paillet, A. Rouxel, H. Carfantan, S. Lacroix, A. Monmayrant, 2025.

[4] A. Wagadarikar, N. P. Pitsianis, X. Sun, D. J. Brady, *Opt. Express* 2009, *17*, 6368, 10.1364/OE.17.006368.

[5] L. Wang, Z. Xiong, D. Gao, G. Shi, W. Zeng, F. Wu, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, p. 4942-4950.

[6] V. Monga, Y. Li, Y. C. Eldar, *IEEE Signal Processing Magazine* 2021, *38*, 18, 10.1109/msp.2020.3016905.

[7] …, J. Lin, H. Wang, X. Yuan, H. Ding, Y. Zhang, R. Timofte, L. V. Gool, in *Advances in Neural Information Processing Systems* (Ed.: S. Koyejo, S. Mohamed,