Introduction to Light Field Analysis

Part II: Non-Lambertian light fields and 3D reconstruction

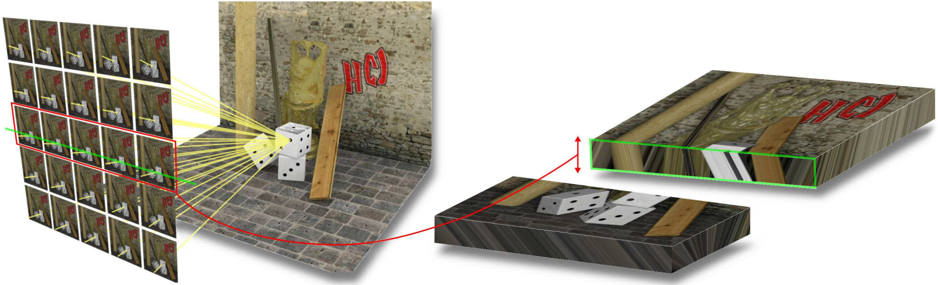
Bastian Goldlücke and Ole Johannsen

GCPR Tutorial
October 2018

computer vision and image analysis

Universität Konstanz

The light field as a dense collection of views



A 2D horizontal cut (green) is called an **epipolar plane image (EPI)**

[Wanner and Goldlücke, CVPR 2012 & TPAMI 2014]

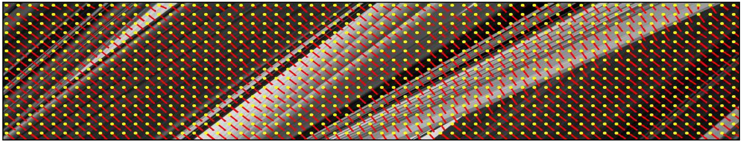
Depth estimation on an epipolar plane image (EPI)

Epipolar plane image (EPI)

[Wanner and Goldlücke, CVPR 2012 & TPAMI 2014]

Depth estimation on an epipolar plane image (EPI)

Epipolar plane image (EPI)

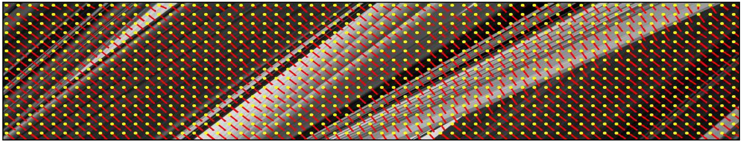


Orientation estimate (structure tensor)

[Wanner and Goldlücke, CVPR 2012 & TPAMI 2014]

Depth estimation on an epipolar plane image (EPI)

Epipolar plane image (EPI)



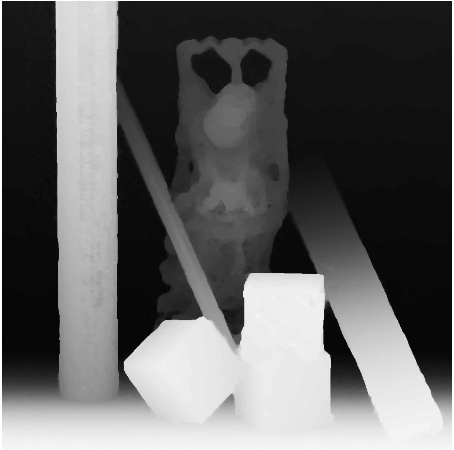
Orientation estimate (structure tensor)

Depth estimate (slope of orientation)

[Wanner and Goldlücke, CVPR 2012 & TPAMI 2014]

Dense depth via orientation estimation

light field center view



estimated depth map (denoised)

[Wanner and Goldluecke CVPR 2012, CVPR 2013, VMV 2013, TPAMI 2014]

Overview

1. Non-Lambertian light fields
   - Multi-layered light fields
   - Sparse coding for multi-layered depth
   - Layer decomposition
   - Specularity removal
   - Intrinsic light field decomposition

2. Structure-from-motion for light field cameras
   - Light field camera pose and alignment
   - Multi-lightfield 3D reconstruction

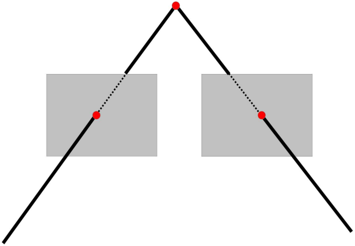
Failure case: “non-cooperative” surfaces

Stereo image pair

Failure case: “non-cooperative” surfaces

Stereo image pair

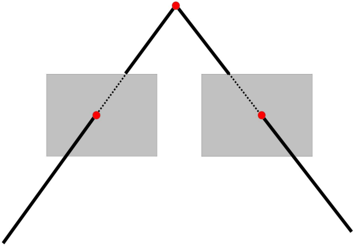
Triangulation from correspondence



Failure case: “non-cooperative” surfaces

Stereo image pair

Triangulation from correspondence



Incorrect assumption: A 3D point looks the same in all views

Failure case: “non-cooperative” surfaces

Stereo image pair

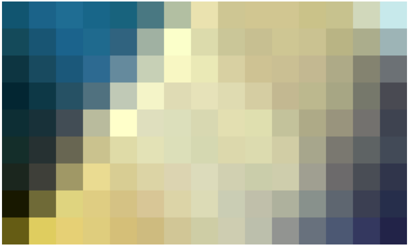
Stereo reconstruction

Incorrect assumption: A 3D point looks the same in all views

Dealing with reflections using the light field

light field

epipolar plane image closeup



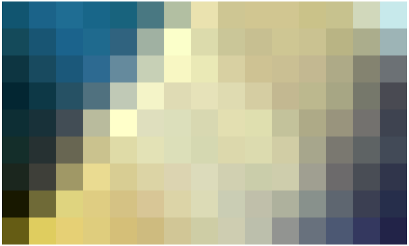
stereo reconstruction

[Wanner and Goldlücke GCPR 2013]

Dealing with reflections using the light field

light field

epipolar plane image closeup



stereo reconstruction

Second order structure tensor:

$$\mathcal{M} = G_\tau * \begin{bmatrix} I_{xx}^2 & I_{xx} I_{xy} & I_{xx} I_{yy} \\ I_{xy} I_{xx} & I_{xy}^2 & I_{xy} I_{yy} \\ I_{yy} I_{xx} & I_{yy} I_{xy} & I_{yy}^2 \end{bmatrix},$$

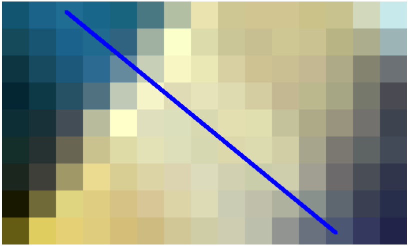
and $\boldsymbol{e}_1(\mathcal{M})$ encodes the two dominant overlaid orientations.

[Wanner and Goldlücke GCPR 2013]

Dealing with reflections using the light field

light field

epipolar plane image closeup



stereo reconstruction

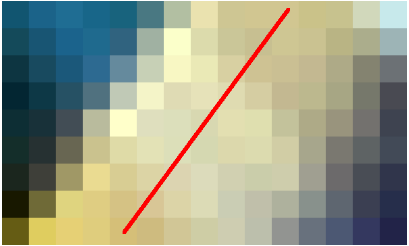
mirror plane depth

[Wanner and Goldlücke GCPR 2013]

Dealing with reflections using the light field

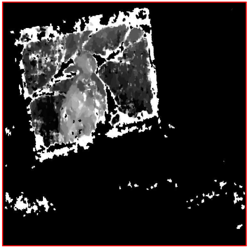
light field

epipolar plane image closeup



stereo reconstruction

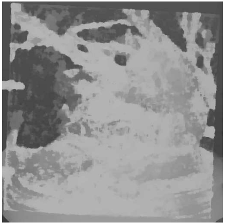
reflection depth

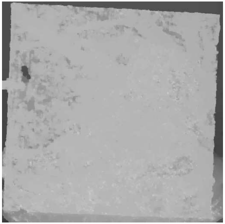


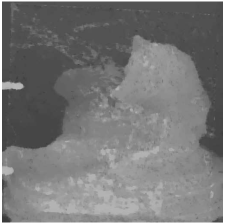
[Wanner and Goldlücke GCPR 2013]

Works great on carefully recorded data ...


light field center view

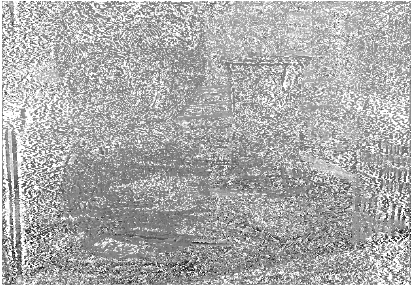

stereo reconstruction


primary surface depth


transmission depth

[Wanner and Goldlücke GCPR 2013]

Sadly, not good enough for plenoptic cameras

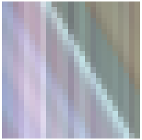


fails due to noise, too large disparity range ...

More robust layered depth reconstruction

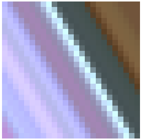
Robust layered depth from sparse light field coding

Idea: represent each EPI patch with atoms of fixed disparity,



$=\quad \alpha_1$

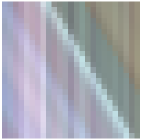
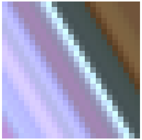
$+\quad \alpha_2$



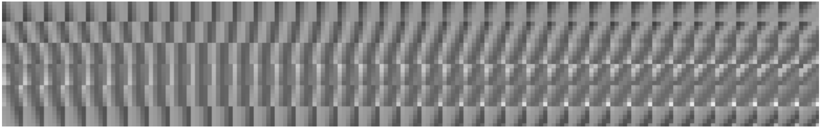
$+\quad \ldots.$

Robust layered depth from sparse light field coding

Idea: represent each EPI patch with atoms of fixed disparity,

 $= \alpha_1$  $+ \alpha_2$  $+ \ldots.$

Each light field patch p is assembled from a trained patch dictionary D

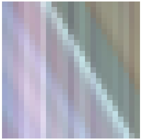
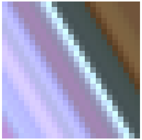


by solving the **sparse coding** problem

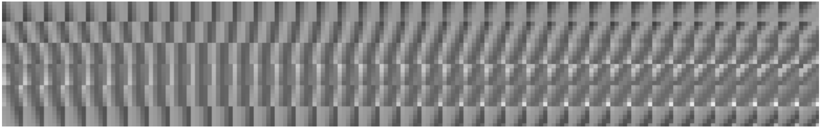
$$\operatorname*{argmin}_{\alpha} \|p - D\alpha\|_2^2 + \lambda \|\alpha\|_1 .$$

Robust layered depth from sparse light field coding

Idea: represent each EPI patch with atoms of fixed disparity,

 $=$ α_1  $+$ α_2  $+ \quad \dots.$

Each light field patch p is assembled from a trained patch dictionary D



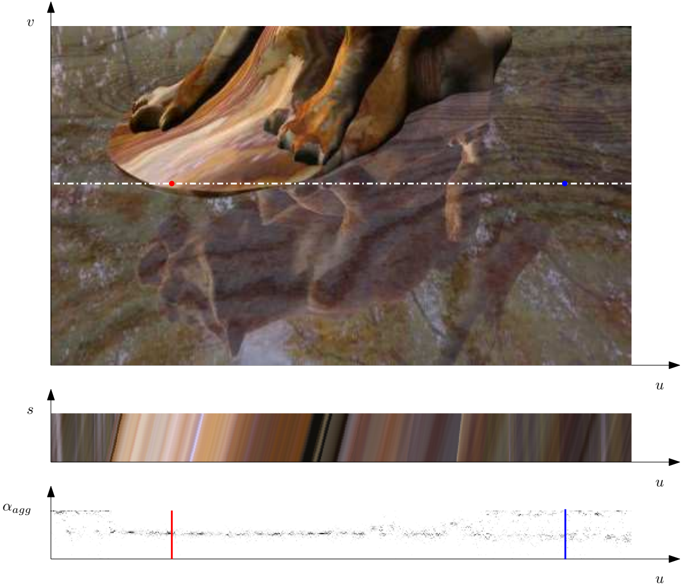
by solving the **sparse coding** problem

$$\underset{\alpha}{\operatorname{argmin}} \|p - D\alpha\|_2^2 + \lambda \|\alpha\|_1 .$$

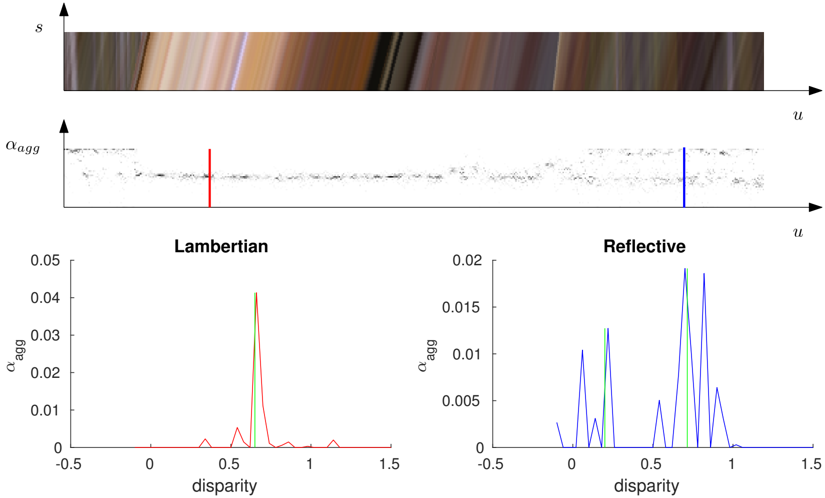
The coding coefficients α should be related to the depth layers.

[Johannsen, Sulc, Goldluecke CVPR 2016]

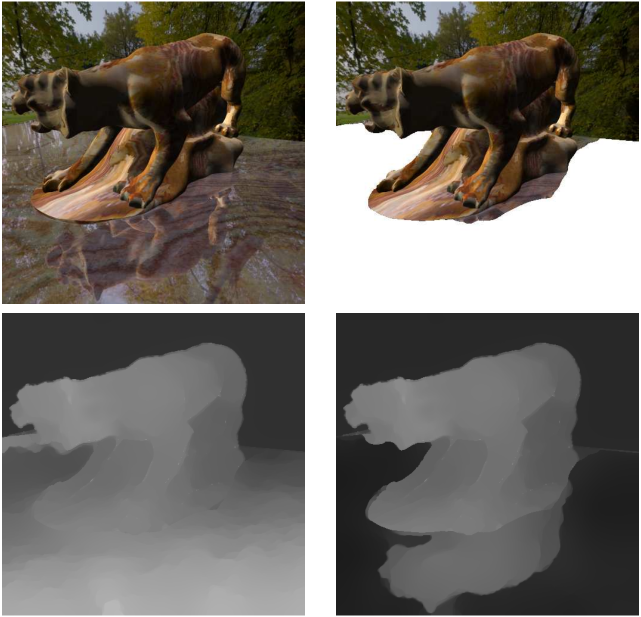
Interpretation of the sparse codes



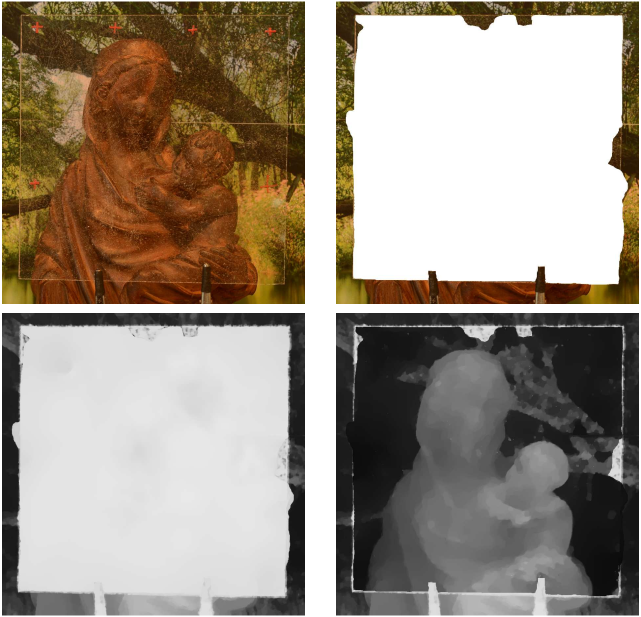
- One peak - Lambertian surface
- Two peaks - Reflective/Transparent surface



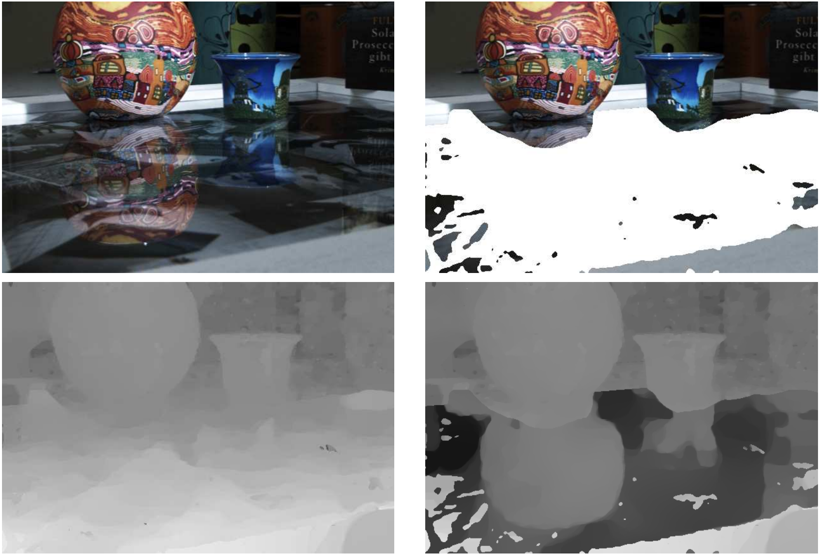
[Johannsen, Sulc, Goldluecke CVPR 2016]



[Johannsen, Sulc, Goldluecke CVPR 2016]



[Johannsen, Sulc, Goldluecke CVPR 2016]

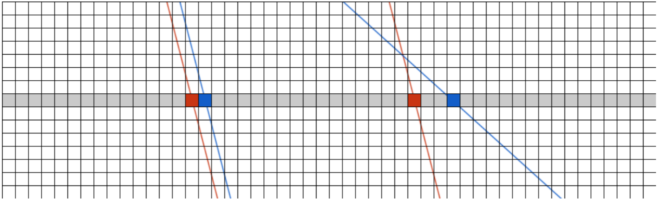


[Johannsen, Sulc, Goldluecke CVPR 2016]

Light field decomposition

[Johannsen, Sulc, Goldluecke VMV 2015]

Generative model: generates EPI from center view



Depth-dependent linear relation between center view and EPI:

$$f = G(d_u)\, u,$$

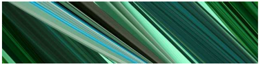
where

u	center view image (at gray pixels),
d_u	center view depth (at gray pixels),
$G(d_u)$	depth-dependent linear transformation,
f	generated complete EPI.

[Johannsen, Sulc, Goldluecke VMV 2015]

Two-layer EPI model

Two-layer EPI synthesis **model**

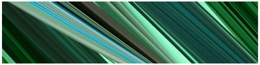


+

=

$G(d_u)u$ $\quad$ $G(d_v)v$ $\quad$ f

Two-layer EPI synthesis **model**

 +  = 

$G(d_u)u$ $G(d_v)v$ f

Leads to **data fitting cost function**

$$D_{EPI}(u,v) = \| G_{d_u} u + G_{d_v} v - f \|_2^2$$

for each individual EPI.

[Johannsen, Sulc, Goldluecke VMV 2015]

- **Dataterm:** sum over all horizontal and vertical EPIs

$$D(u, v) = \sum_{x=1}^{W} D_x(u_x, v_x) + \sum_{y=1}^{H} D_y(u_y, v_y).$$

Variational inverse problem for layer decomposition

- **Dataterm:** sum over all horizontal and vertical EPIs

$$D(u,v) = \sum_{x=1}^{W} D_x(u_x, v_x) + \sum_{y=1}^{H} D_y(u_y, v_y).$$

- **Regularization:** total generalized variation (TGV_2), favors piecewise linear solutions

$$J(u,v) = \mathrm{TGV}_2(u) + \mathrm{TGV}_2(v).$$

Variational inverse problem for layer decomposition

- **Dataterm:** sum over all horizontal and vertical EPIs

$$D(u,v) = \sum_{x=1}^{W} D_x(u_x, v_x) + \sum_{y=1}^{H} D_y(u_y, v_y).$$

- **Regularization:** total generalized variation (TGV_2), favors piecewise linear solutions

$$J(u,v) = \mathrm{TGV}_2(u) + \mathrm{TGV}_2(v).$$

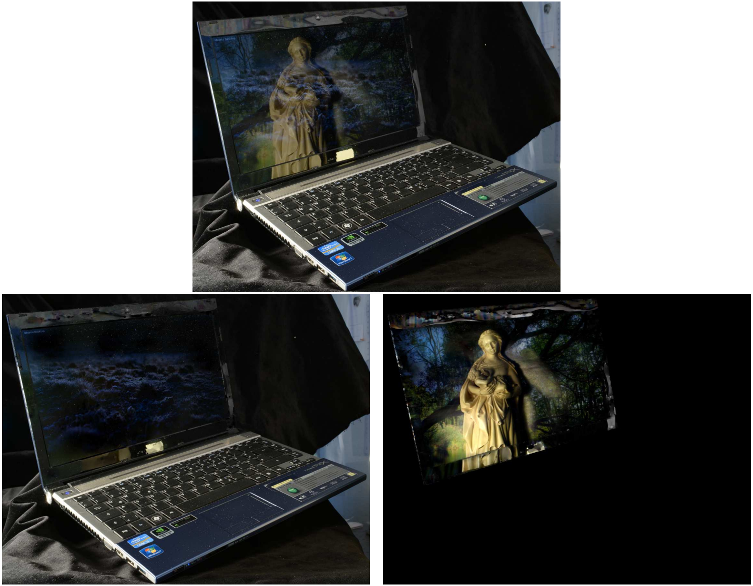
- **Total energy**:

$$E(u,v) = D(u,v) + \lambda J(u,v)$$

minimize e.g. with primal-dual method [Chambolle and Pock 2010].

[Johannsen, Sulc, Goldluecke VMV 2015]

[Johannsen, Sulc, Goldluecke VMV 2015]



[Johannsen, Sulc, Goldluecke VMV 2015]

Specularity removal

Can we get rid of the specular reflections?

Very simple model to get started

Light field is composed of a sparse set of diffuse albedos and a specular color, color in each ray a sparse linear combination

$$L(\boldsymbol{r}) = \alpha_1(\boldsymbol{r})C_1 + \cdots + \alpha_k(\boldsymbol{r})C_k + \sigma(\boldsymbol{r})S$$

Optimize for dictionary D of colors and the coefficients in

$$E(D, \boldsymbol{\alpha}, s) = \left\| L - D \begin{bmatrix} \boldsymbol{\alpha} \\ \sigma \end{bmatrix} \right\|_2^2 + \lambda \left\| \boldsymbol{\alpha} \right\|_1 + J(\boldsymbol{\alpha}) + J(\sigma).$$

For single view, this was proposed by *[Akashi et al. 2014]*.

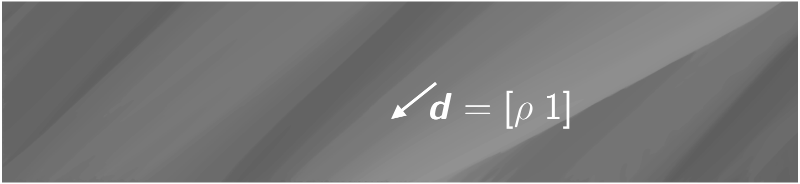
The regularization of the diffuse coefficients must follow disparity as before. The specular component, however, stays constant along the **specular flow $\boldsymbol{w}$**, which relates motion of speular highlights to camera motion *[Blake et al. 1991, Adato et al. 2007]*.

- Complete problem is 4D - too large to handle all at once.
- Regularizer separated into **independent 2D components** on epipolar plane images in (y, t) and (x, s) coordinates, as well as pinhole views in (x, y) coordinates:

$$
\begin{aligned}
J(\boldsymbol{U}) = &\int J_{\text{epi}}(\boldsymbol{U}_{xs})\,\mathrm{d}(x, s) \\
&+ \int J_{\text{epi}}(\boldsymbol{U}_{yt})\,\mathrm{d}(y, t) \\
&+ \int J_{\text{view}}(\boldsymbol{U}_{st})\,\mathrm{d}(s, t).
\end{aligned}
$$

Goldlücke and Wanner CVPR 2013

Regularization in the **direction of epipolar lines** given by the disparity field ρ:

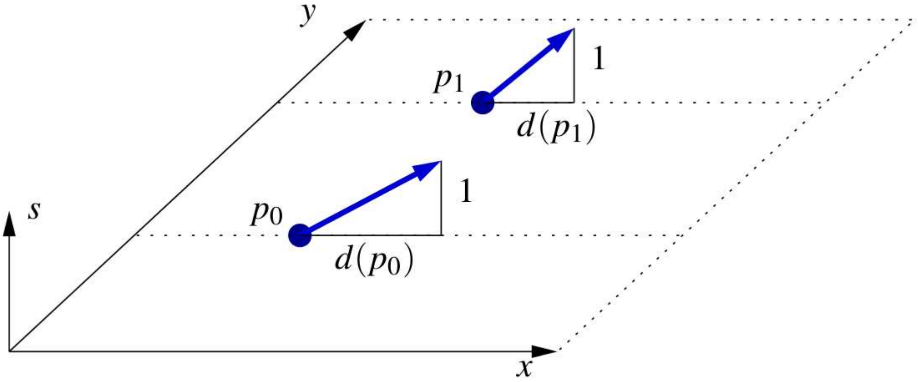


Achieved by *anisotropic total variation*

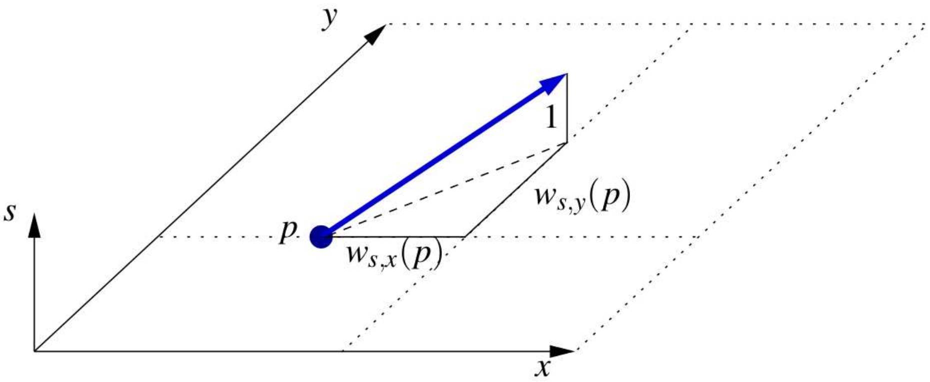
$$J_{\mathrm{epi}}(\boldsymbol{U}_{yt}) := \sum_{i=1}^{d} \int \sqrt{(\nabla U_{yt}^i)^T D_\rho \nabla U_{yt}^i}\, \mathrm{d}(x, s),$$

tensor D_ρ encodes direction information.

Goldlücke and Wanner CVPR 2013

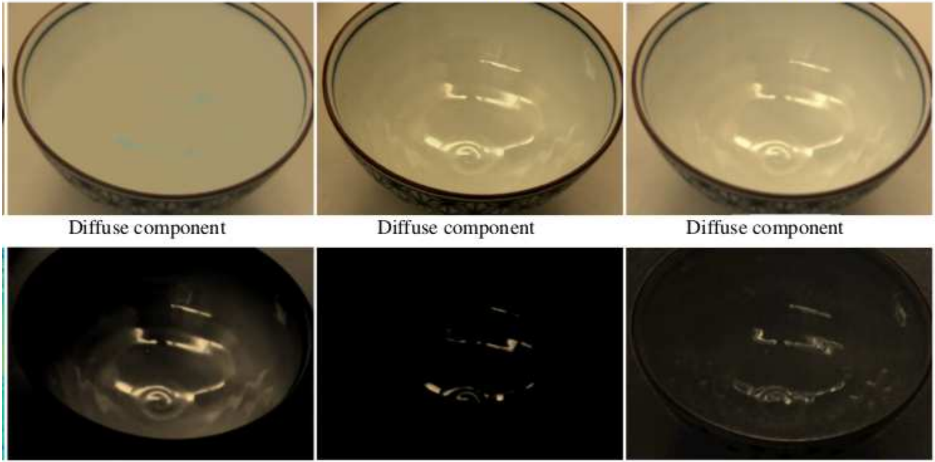

y
1
p1
d(p1)
1
s
p0
d(p0)
x

Epipolar volume regularization for specular components



Results

Our result	Single view [AO14]	Tao et al. [TSW*15]

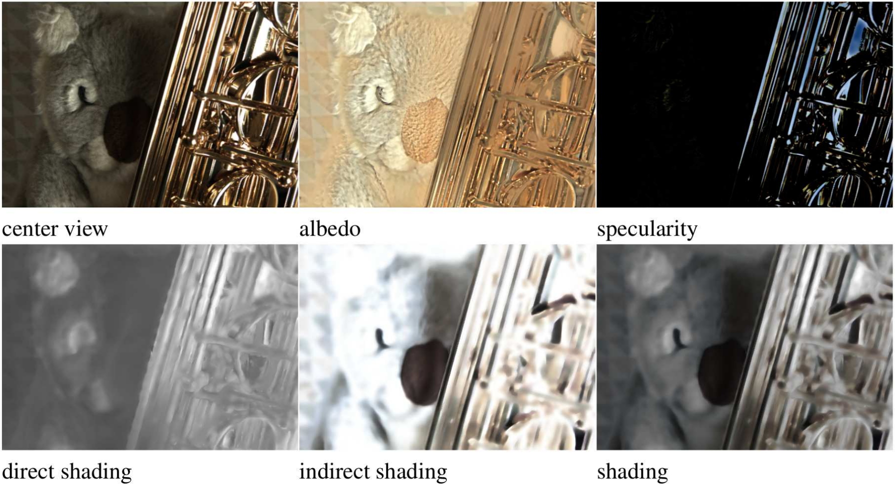


Diffuse component — Diffuse component — Diffuse component

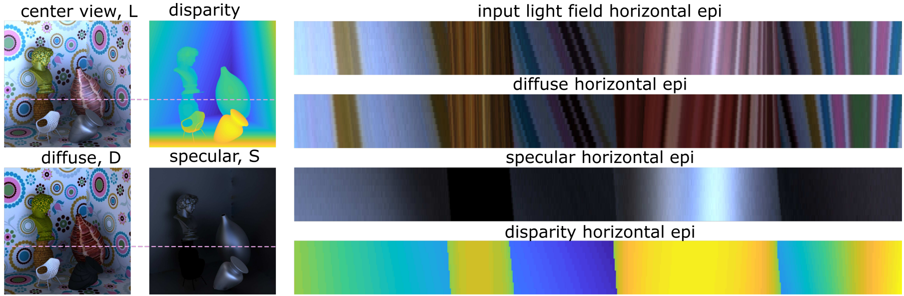
Specular component — Specular component — Specular component

Intrinsic light field decomposition

Intrinsic light field decomposition



Dichromatic Reflection model



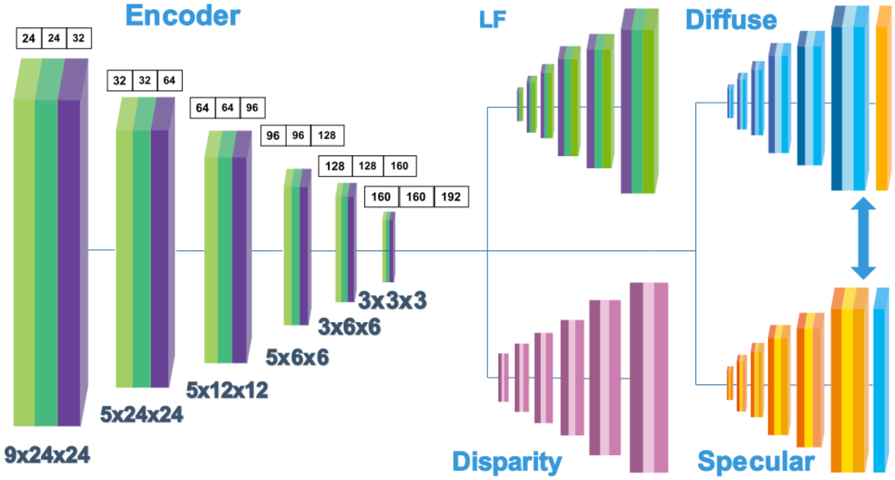
- Input light field L can be additively separated into its diffuse and specular components D and S,

$$L = D + S$$

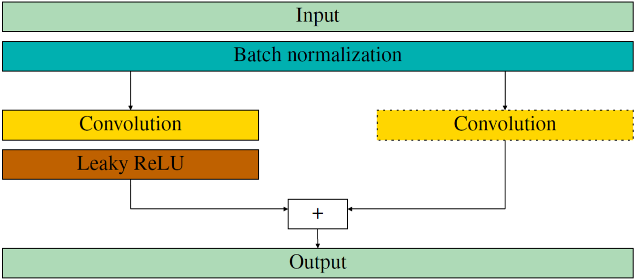
.

- **Diffuse component** and the disparity correspond to the same projections of the same 3D points and share the same pattern.
- **Specular component** follows the specular flow, which depends on the local surface geometry and view point change.

Neural network: one encoder, several decoders

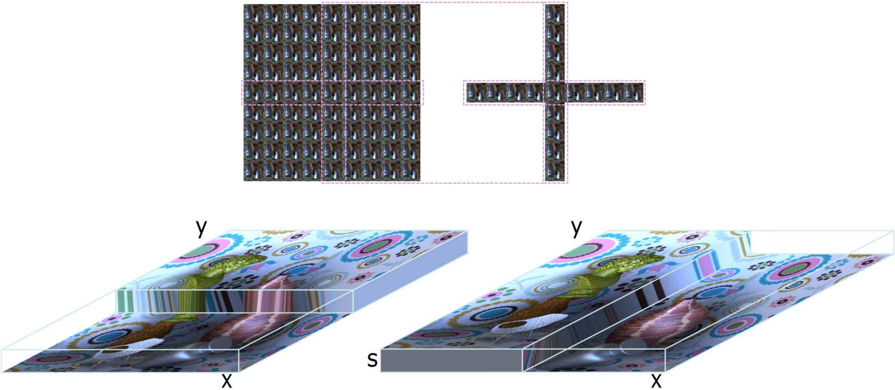


Single residual layer



- After **batch normalization**, a first path leads through a (possibly strided) **convolution** layer and a **leaky ReLU**.
- A second path either keeps the input, or passes it through a strided **convolution** in case it needs to be resampled.
- Both paths are added together to produce the final output.

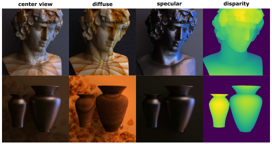
Network inputs



- crosshair-shaped subset of 17 views
- patch size $9 \times 48 \times 48$
- $\approx 160,000$ patches per light field

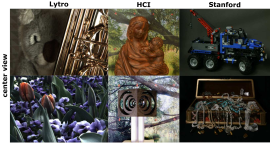
Training and testing data

Synthetic data



Light field benchmark

Real world data

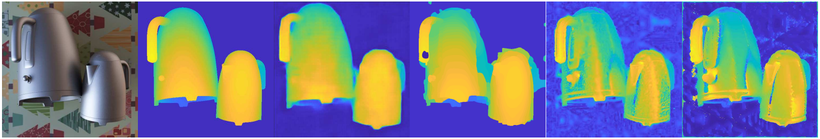


Additional synthetic data: generated based on Blender plugin provided with the benchmark.

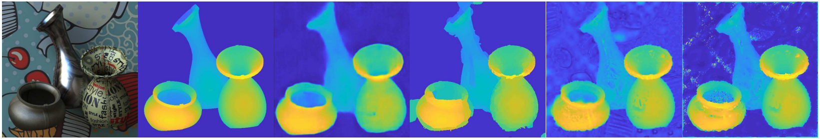
- 171 scenes, 321 textures, 109 environmental maps[3]
- 36 pre-built scenes with objects from Chocofur[1] and The British Museum[2]
- Randomized position, amount of specularity, texture and color

Comparisons: disparity

input	gt	**Ours**	ACC [1]	EPI1 [2]	EPI2 [3]



	MSE $\times 100$:	**5.9**	23.1	30.0	35.5



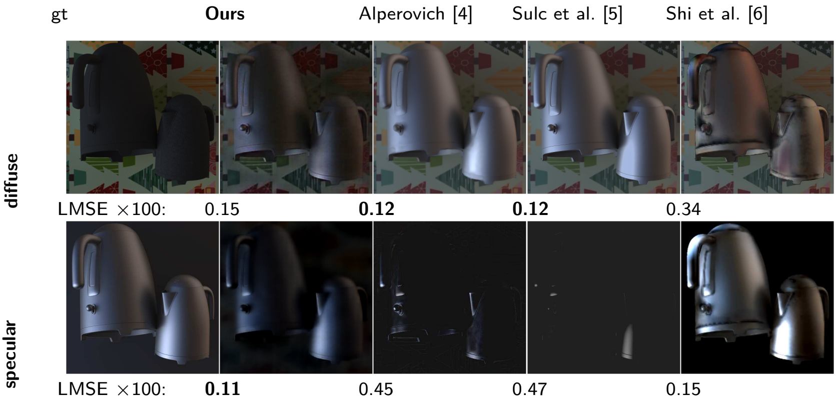
	MSE $\times 100$:	**4.6**	13.4	14.7	17.3

[1] H. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y. Tai, and I. Kweon. Accurate depth map estimation from a lenslet light field camera. In *CVPR*, 2015.

[2] O. Johannsen, A. Sulc, and B. Goldluecke. What sparse light field coding reveals about scene structure. In *CVPR*, 2016.

[3] S. Wanner and B. Goldluecke. Globally consistent depth labeling of 4D light fields. In *CVPR*, 2012.

Comparisons: reflection separation



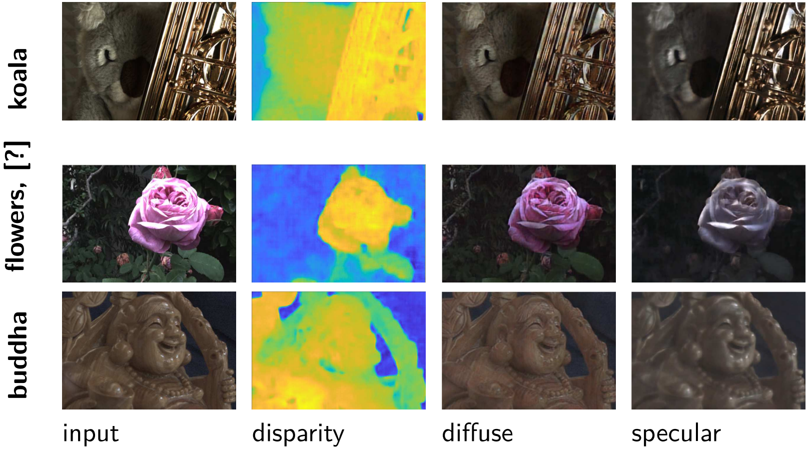
	Ours	Alperovich [4]	Sulc et al. [5]	Shi et al. [6]
diffuse LMSE $\times 100$:	0.15	**0.12**	**0.12**	0.34
specular LMSE $\times 100$:	**0.11**	0.45	0.47	0.15

[4] A. Alperovich, O. Johannsen, M. Strecke, and B. Goldluecke. Shadow and specularity priors for intrinsic light field decomposition. In *EMMCVPR*, 2017.

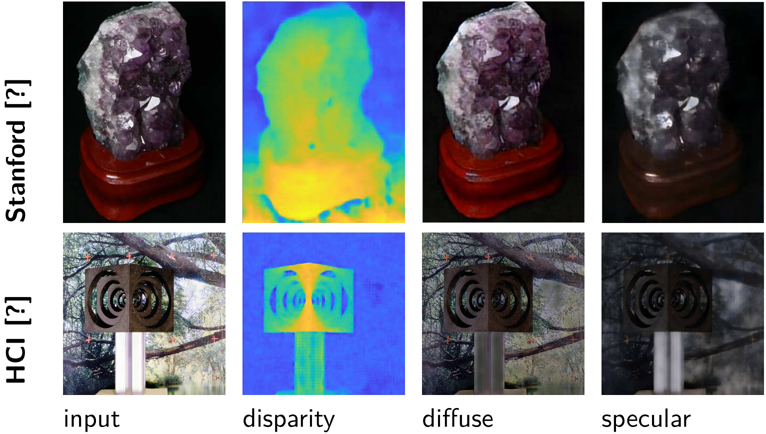
[5] A. Sulc, A. Alperovich, N. Marniok, and B. Goldluecke. Reflection separation in light fields based on sparse coding and specular flow. In *VMV*, 2016.

[6] J. Shi, Y. Dong, H. Su, and S. Yu. Learning non-lambertian object intrinsics across shapenet categories. In *CVPR*, 2017.

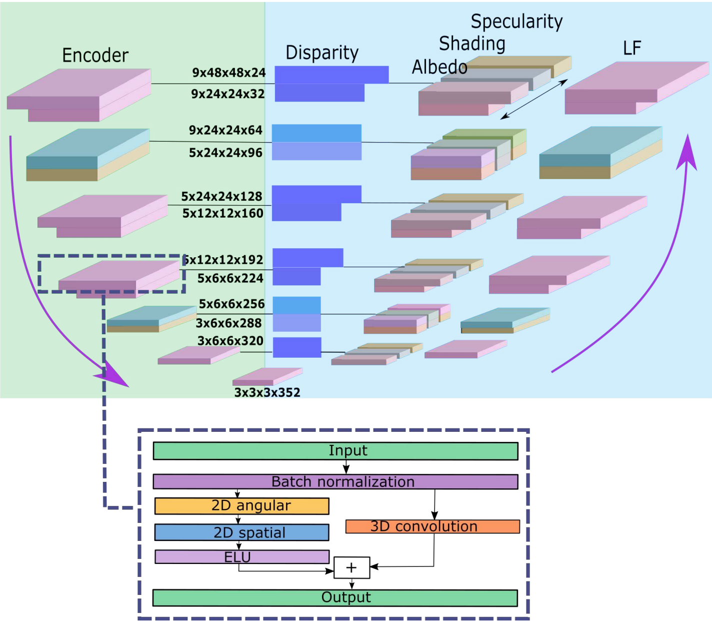
Results: Lytro



Results: gantry datasets



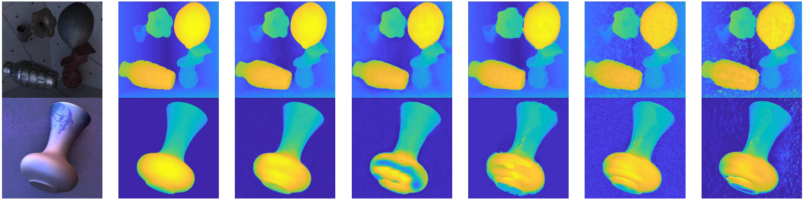
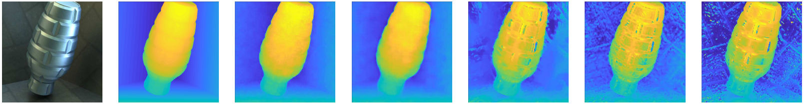
Extension: recovering shading

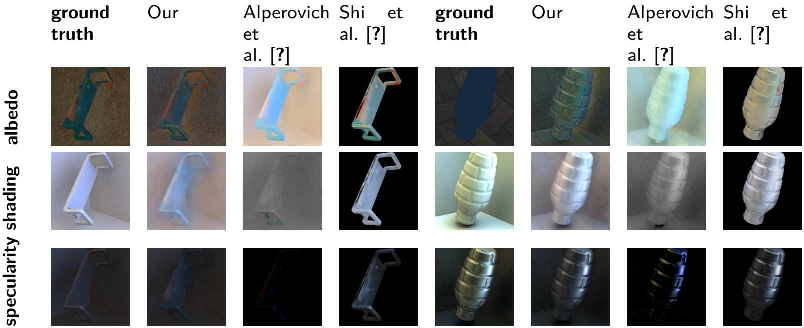


Code and datasets available

`https://github.com/cvia-kn/`

Comparison: disparity

center view	**ground truth**	**Ours**	Alperovich et al.	Strecke et al.	Johannsen et al.	Wanner et al.
	$MSE \times 100$	**2.2**	**30.25**	**12.02**	**9.53**	**15.64**
	$MSE \times 100$	**0.52**	**1.57**	**7.87**	**8.19**	**18.56**



Comparison on two synthetic data sets generated with Blender. The light field size is $9 \times 9 \times 512 \times 512 \times 3$. We compare with the modeling approach for light fields [?] and single image CNN [?].

Results: Lytro

center view **albedo** **shading** **specularity**

Overview

1. Non-Lambertian light fields
   - Multi-layered light fields
   - Sparse coding for multi-layered depth
   - Layer decomposition
   - Specularity removal
   - Intrinsic light field decomposition

2. Structure-from-motion for light field cameras
   - Light field camera pose and alignment
   - Multi-lightfield 3D reconstruction

Light field camera pose and alignment

[Johannsen, Sulc, Goldluecke ICCV 2015]

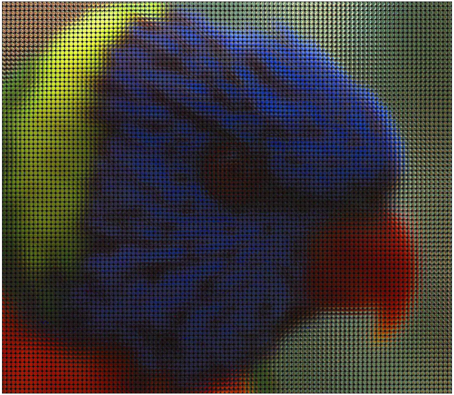
[Johannsen, Sulc, Goldluecke ICCV 2015]

Research questions

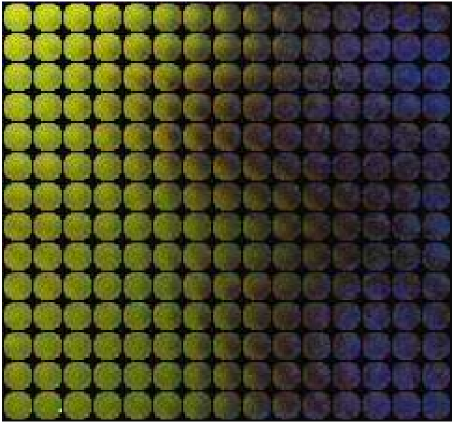
- How to estimate pose for light field cameras?
- How to easily align light fields for panoramas?
  - sparse correspondence only
  - tailored to light field geometry
  - linear algorithm

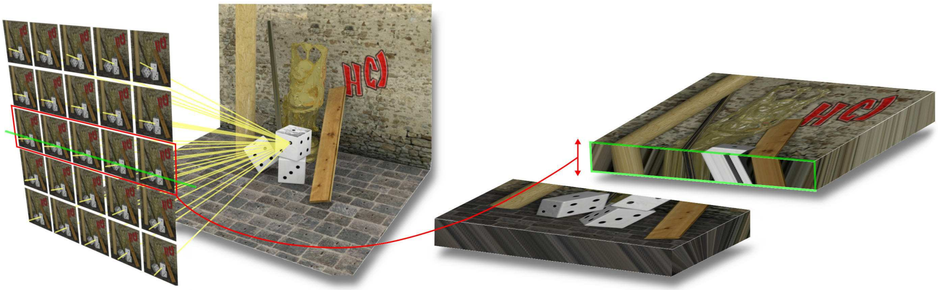
- Input: light field from pre-calibrated plenoptic camera (i.e. raw image decoded into two-plane parametrization).

Lytro camera and microlens images



Lytro camera and microlens images

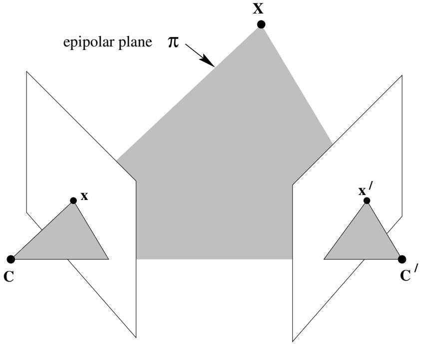




... but evil non-linear distortions, out of scope of this tutorial. In practice: use Matlab calibration toolbox provided by Donald Dansereau,

`http://dgd.vision/Tools/LFToolbox/`

Reminder: SfM for conventional cameras



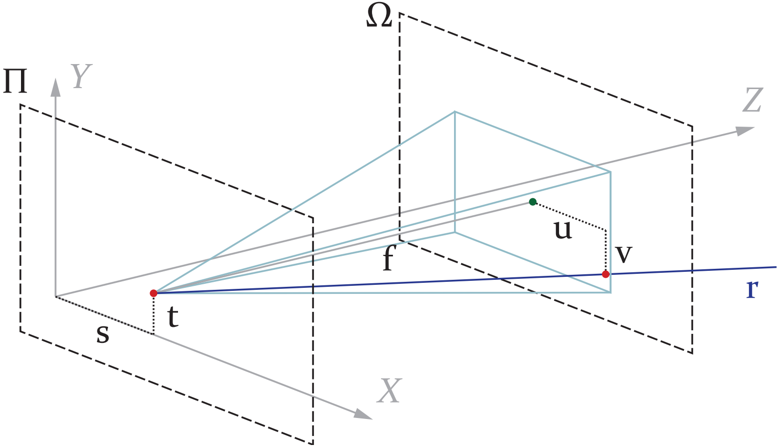
- Correspondence constraints $(\boldsymbol{x}')^T F \boldsymbol{x} = 0$ with **Fundamental Matrix** F
- Related to projections P, P', rotation R and translation t between views,

$$F = \left[-P' R^T t\right]_\times P' P^+.$$

- Allows to recover rotation and translation ("pose") if cameras are fully calibrated.

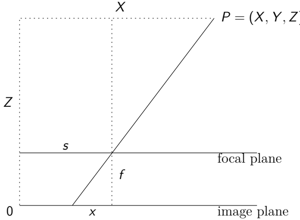
Figures: Hartley and Zisserman

Light field geometry



What is the projection of a 3D point $\boldsymbol{X}$ into a light field?

The slope-depth relationship



Intercept theorem (pinhole perspective projection):

$$\frac{x}{f} = \frac{(X - s)}{Z}, \qquad \frac{y}{f} = \frac{Y - t}{Z}.$$

The projection coordinates for two different subaperture views (s_1, t_1), (s_2, t_2) satisfy

$$x_2 - x_1 = -\frac{f}{Z}(s_2 - s_1), \qquad y_2 - y_1 = -\frac{f}{Z}(t_2 - t_1).$$

Result: for a given depth (distance) Z of a scene point to the focal plane, there is a linear relationship between projection and view point coordinates. The "scale factor" $d = \frac{f}{Z}$ is called **disparity**.

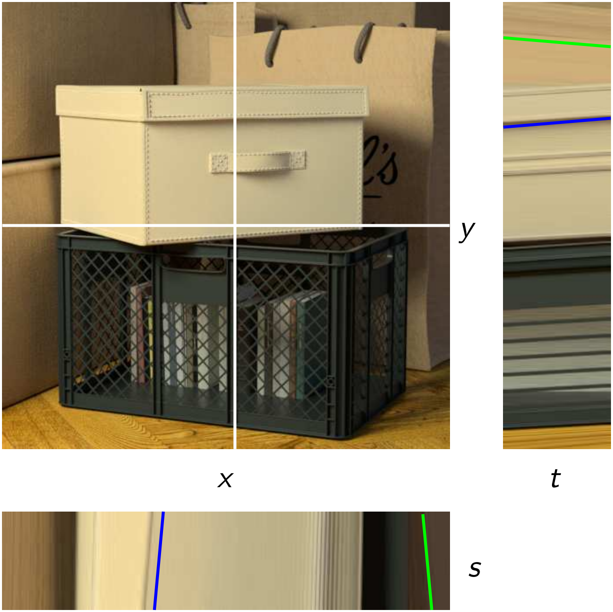
What is the projection of a 3D point $\boldsymbol{X}$ into a light field?

A 2D subspace in the 4D ray space, which can be parametrized in 5D homogenous light field coordinates as follows:

$$\underbrace{\begin{bmatrix} 1 & 0 & \frac{f}{Z} & 0 & -\frac{fX}{Z} \\ 0 & 1 & 0 & \frac{f}{Z} & -\frac{fY}{Z} \end{bmatrix}}_{=:M(\boldsymbol{X},f)} \begin{bmatrix} u \\ v \\ s \\ t \\ 1 \end{bmatrix} = 0.$$

Note: these are just the projection equations of a pinhole camera located in (s, t).

Epipolar plane images (EPIs)



What are correspondences between two light fields?

What are correspondences between two light fields?

- Ideally, correspondences between subspaces.

What are correspondences between two light fields?

- Ideally, correspondences between subspaces.
- In practice, matched features between subaperture images:

$$\{\boldsymbol{r}_i\}_{i=1,\ldots n} \leftrightarrow \{\boldsymbol{r}'_j\}_{j=1,\ldots,m}.$$

where $\boldsymbol{r}, \boldsymbol{r}'$ are 4D light field coordinates in two different light fields - i.e. rays.

What are correspondences between two light fields?

- Ideally, correspondences between subspaces.
- In practice, matched features between subaperture images:

$$\{\boldsymbol{r}_i\}_{i=1,\ldots n} \leftrightarrow \{\boldsymbol{r}'_j\}_{j=1,\ldots,m}.$$

  where $\boldsymbol{r}, \boldsymbol{r}'$ are 4D light field coordinates in two different light fields - i.e. rays.

- obtained e.g. from matching SIFT features across all subaperture images, absurd matches pre-eliminated (if e.g. disparity outside a sensible range).

- **Obvious observation**: a single LF correspondence leads to two corresponding 3D points.

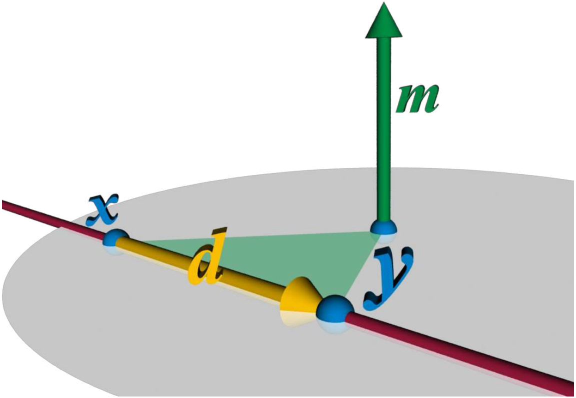
- **Obvious observation**: a single LF correspondence leads to two corresponding 3D points.

- Trivial strategy for pose estimation: compute 3D points for all LHS and RHS of a correspondence, align two corresponding 3D point clouds.

Light field camera pose: strategy I

- **Obvious observation**: a single LF correspondence leads to two corresponding 3D points.
- Trivial strategy for pose estimation: compute 3D points for all LHS and RHS of a correspondence, align two corresponding 3D point clouds.
- Turns out this is too simple and does not work very well.

Rays in Plücker coordinates



From two points x, y on the line:
Displacement $d = y - x$, momentum $m = x \times y$.
Pair $[d; m]$ is an invariant for the line (up to scale).
Displacement denoted with $\boldsymbol{q}$ on followup slides.

Image: Wikipedia

Light field camera pose: strategy II

- Consider two rays $\boldsymbol{r} = [\boldsymbol{q}; \boldsymbol{m}]$ and $\boldsymbol{r}' = [\boldsymbol{q}'; \boldsymbol{m}']$ which intersect in a common point.

- Generalized epipolar constraint:

$$\boldsymbol{q}'^{T} E \boldsymbol{q} + \boldsymbol{q}'^{T} R \boldsymbol{m} + \boldsymbol{m}'^{T} R \boldsymbol{q} = 0.$$

  where $E = [t]_\times R$ is the essential matrix, and the camera coordinate systems are related by a rotation R and translation t according to

$$\boldsymbol{X}' = R\boldsymbol{X} + t.$$

- Allows to recover pose from corresponding pairs of rays.

- Note number of equations: $n \cdot m$ per light field correspondence.

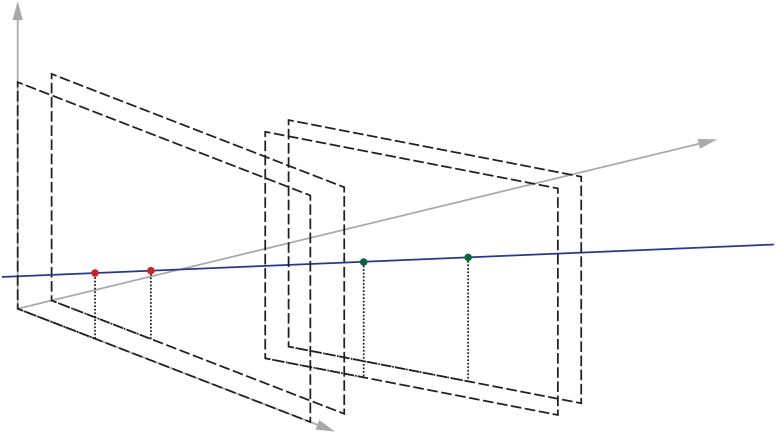
Linear correspondence constraints in light fields?

Linear correspondence constraints in light fields?

Observation I: projection from Plücker ray coordinates into homogenous light field coordinates is projective-linear:

$$q_3' \begin{bmatrix} u' \\ v' \\ s' \\ t' \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 & 0 & 0 \\ 0 & f & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} R & 0 \\ E & R \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{m} \end{bmatrix}.$$

Observation II: Each ray in a correspondence must lie in the subspace of the correspondence when transformed into the respective other light field.



Light field camera pose: proposed strategy

Given a correspondence

$$\{\boldsymbol{r}_i\}_{i=1,\dots n} \leftrightarrow \{\boldsymbol{r}'_j\}_{j=1,\dots,m}$$

estimate subspace matrices M and M' for LHS and RHS.

Given a correspondence

$$\{\boldsymbol{r}_i\}_{i=1,\dots n} \leftrightarrow \{\boldsymbol{r}'_j\}_{j=1,\dots,m}$$

estimate subspace matrices M and M' for LHS and RHS.

Each ray on the LHS must satisfy

$$M' P(f) \begin{bmatrix} R & 0 \\ E & R \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{m} \end{bmatrix} = 0.$$

Light field camera pose: proposed strategy

Given a correspondence

$$\{\boldsymbol{r}_i\}_{i=1,\dots n} \leftrightarrow \{\boldsymbol{r}'_j\}_{j=1,\dots,m}$$

estimate subspace matrices M and M' for LHS and RHS.

Each ray on the LHS must satisfy

$$M'P(f)\begin{bmatrix} R & 0 \\ E & R \end{bmatrix}\begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{m} \end{bmatrix} = 0.$$

Abbreviate with M_1 the first three and with M_2 the second three columns of the 2×6 matrix $M'P(f)$,

$$M_1 R\boldsymbol{q} + M_2 R\boldsymbol{m} + M_1 E\boldsymbol{q} = 0.$$

Same form as GEC, same algorithm to compute solution.
Note: only $2(n+m)$ equations per correspondence instead of $n \cdot m$.

Can be re-arranged to

$$A_E \mathrm{vec}(E) + A_R \mathrm{vec}(R) \ = \ 0.$$

with matrices E, R stacked to columns $\mathrm{vec}(E), \mathrm{vec}(R)$.

Can be re-arranged to

$$A_E \text{vec}(E) + A_R \text{vec}(R) \ = \ 0.$$

with matrices E, R stacked to columns $\text{vec}(E), \text{vec}(R)$.

Solution for $\text{vec}(R)$ s.t. $\|\text{vec}(R)\| = 1$ satisfies

$$(A_E A_E^+ - I) A_R \text{vec}(R) = \mathbf{0}.$$

Solve using SVD, project to $SO(3)$ to obtain linear estimate for R.

Can be re-arranged to

$$A_E \mathrm{vec}(E) + A_R \mathrm{vec}(R) \; = \; 0.$$

with matrices E, R stacked to columns $\mathrm{vec}(E), \mathrm{vec}(R)$.

Solution for $\mathrm{vec}(R)$ s.t. $\|\mathrm{vec}(R)\| = 1$ satisfies

$$(A_E A_E^+ - I) A_R \mathrm{vec}(R) = \mathbf{0}.$$

Solve using SVD, project to $SO(3)$ to obtain linear estimate for R.

Linear estimate for t follows from substituting solution for R into above equation.

In theory, solution is subject to difficult non-linear constraints:

- R must be a rotation,
- $E = [t]_\times R$,

so it is not expected that the linear estimate is sufficient.

Non-linear refinement

In theory, solution is subject to difficult non-linear constraints:

- R must be a rotation,
- $E = [t]_\times R$,

so it is not expected that the linear estimate is sufficient.

Previous work: "iterative refinement", solve for R and t in turn, backproject to allowed space of solutions.

Non-linear refinement

In theory, solution is subject to difficult non-linear constraints:

- R must be a rotation,
- $E = [t]_\times R$,

so it is not expected that the linear estimate is sufficient.

Previous work: "iterative refinement", solve for R and t in turn, backproject to allowed space of solutions.

In practice, not necessary **if solving for R first instead of E**.

Focal length

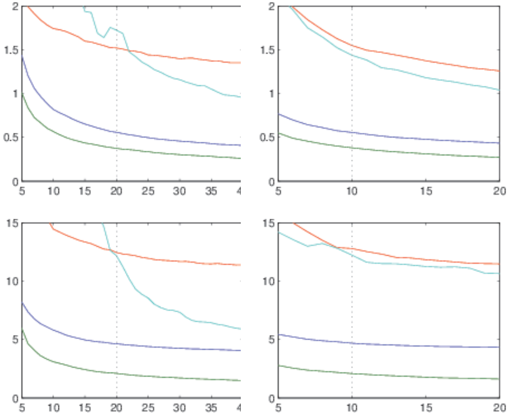
Can be done by brute force search, just look for f which minimizes residual in the linear system - not elegant, but works. Full non-linear bundle adjustment as a second step of course possible as well.

Light field camera pose: comparison

	Correspondences	10 matches, 10 projections per point					20 matches, 10 projections per point					10 matches, 20 projections per point				
	Noise level σ_{uv}	0.2	0.4	0.6	0.8	1.0	0.2	0.4	0.6	0.8	1.0	0.2	0.4	0.6	0.8	1.0
Angular rot. error [deg]	*linear methods*															
	3DPC	1.82	5.19	8.93	11.56	16.18	1.33	3.42	6.77	8.38	10.24	1.15	4.35	6.43	9.14	14.12
	R2R-O	2.04	11.17	15.37	37.48	43.32	0.72	2.19	3.48	3.76	6.41	1.87	3.01	17.56	39.66	40.02
	R2R-I	0.88	1.91	3.51	4.55	5.19	0.40	0.92	2.13	2.76	4.01	0.55	1.40	2.64	4.20	5.98
	Proposed	**0.65**	**1.19**	1.80	2.28	3.15	0.27	0.52	0.83	1.11	1.49	0.40	0.81	1.27	1.77	**2.39**
	with refinement															
	R2R-O-R20	1.00	2.05	3.71	4.53	18.40	0.43	0.90	2.31	2.94	4.10	0.62	1.45	9.27	7.86	9.33
	R2R-I-R20	1.00	2.18	3.83	4.73	5.21	0.43	0.91	2.32	2.95	4.11	0.64	1.50	2.74	4.45	6.08
	Proposed-R20	0.69	1.20	**1.77**	**2.23**	**2.85**	**0.26**	**0.50**	**0.79**	**1.05**	**1.42**	**0.37**	**0.80**	**1.18**	**1.57**	2.39
Relative transl. error [%]	*linear methods*															
	3DPC	0.28	0.82	0.93	2.19	2.20	0.32	0.97	0.92	1.44	0.95	0.46	0.84	1.21	0.74	0.86
	R2R-O	0.20	1.15	1.06	4.18	2.01	0.05	0.43	0.30	0.36	0.45	1.68	0.27	1.43	0.98	0.79
	R2R-I	0.04	0.12	0.23	0.46	0.52	0.02	0.09	0.14	0.25	0.36	0.12	0.11	0.27	0.24	0.33
	Proposed	**0.03**	**0.07**	0.15	**0.25**	0.24	**0.01**	0.05	**0.07**	**0.11**	0.14	0.13	**0.06**	0.16	0.10	**0.12**
	with refinement															
	R2R-O-R20	0.05	0.13	0.24	0.51	0.67	0.02	0.09	0.15	0.25	0.36	0.12	0.11	0.41	0.27	0.37
	R2R-I-R20	0.04	0.13	0.24	0.51	0.51	0.02	0.09	0.15	0.25	0.36	0.12	0.11	0.27	0.25	0.33
	Proposed-R20	0.03	0.08	**0.14**	0.26	**0.23**	0.01	**0.05**	0.07	0.11	**0.14**	**0.11**	0.06	**0.16**	**0.10**	0.13

Accuracy of the different methods both before and after non-linear refinement. Different numbers of correspondences N, projections per correspondence K, and levels of noise σ_{uv} on the (u, v)-coordinates are compared. Error metrics are the angular deviation from the ground truth in degrees for the estimated rotation, as well as the relative translation error measured as a percentage of the length of the ground truth translation vector. Noise standard deviation is given in units of pixels on the subaperture images. In all cases, the most accurate method (highlighted in bold) is the one proposed in this paper.

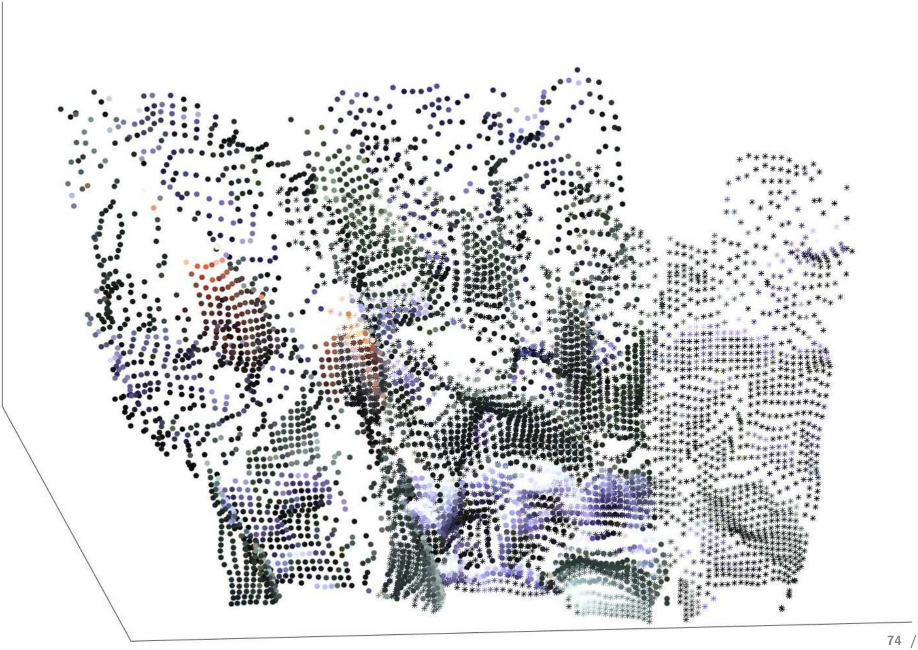
Accuracy over number of corresponding rays



The graphs show how the angular error in rotation depends on the number of matches (left) and the number of rays per match (right). Compared are the four linear methods in table **??**: 3DPC [**?**] (red) and R2R-O [**?**] (cyan), R2R-I with our proposed numerical improvements (blue), and finally the novel proposed method for 4D light fields (green). Top row: small amount of noise ($\sigma = 0.2$), bottom: large amount of noise ($\sigma = 1.0$).

Living panoramas

3D point cloud



Conclusions

- Simple linear method to estimate pose for light fields in the two-plane parametrization.

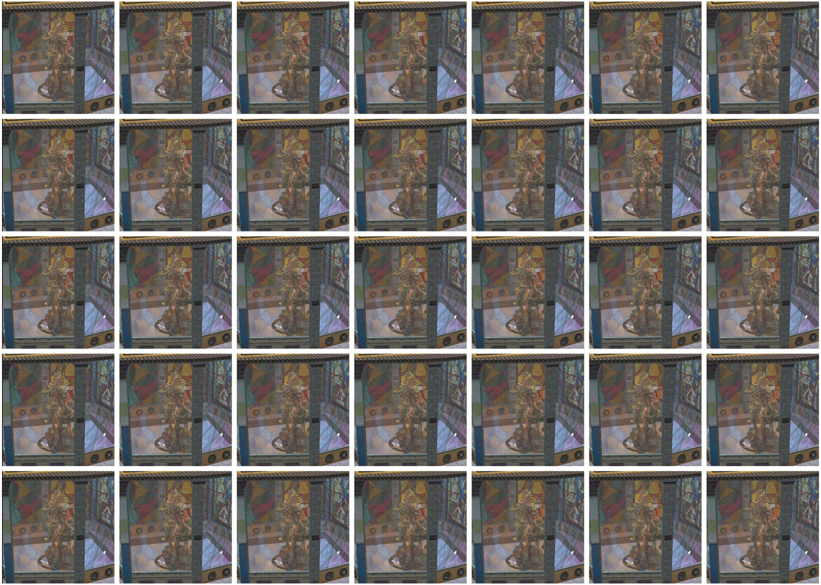
- More accurate than all previous methods, reduced number of equations compared to framework of generalized cameras.

- Allows simple construction of refocusable light field panoramas, but there's work left to do for high quality.

Putting it all together - full scene reconstruction

What do these images tell you about the scene?

The light field: densely sampled view points

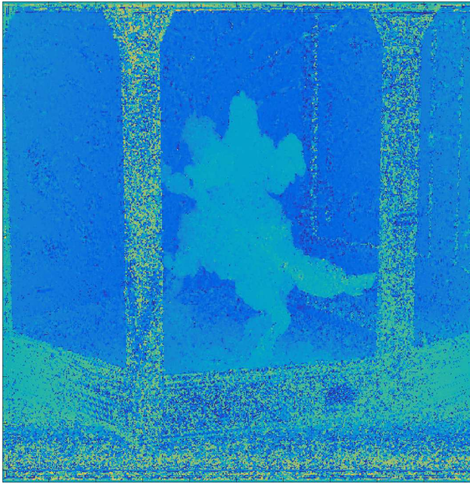


What if each of these views is actually a light field?

Let's go back to our challenge dataset

Two-layered depth map estimation

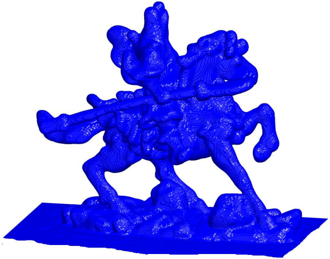
Center view 14 / 24



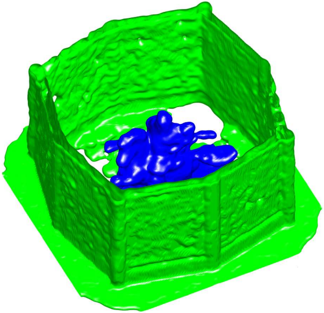
Depth second layer

Input

reconstruction



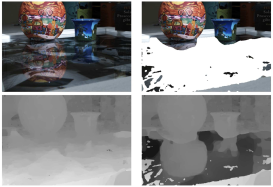
ground truth

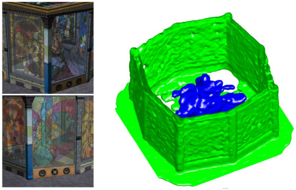


Summary

Sparse light field coding for multi-layer depth

[Johannsen, Sulc, Goldluecke CVPR 2016, GCPR 2016]





Multi-layered 3D scene reconstruction

[Johannsen, Sulc, Marniok, Goldluecke ICCV 2015, ACCV 2016],

Light field decomposition and intrinsic light fields

[Johannsen, Sulc, Goldluecke VMV 2015, 2016
Alperovich and Goldluecke ACCV 2016, EMMCVPR 2017,
Alperovich, Johannsen, Strecke, Goldluecke CVPR 2018]